

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平10-135849

(43)公開日 平成10年(1998) 5月22日

(51)Int.Cl. ⁶	識別記号	F I	
H 0 3 M 13/00		H 0 3 M 13/00	
G 0 6 F 11/10	3 3 0	G 0 6 F 11/10	3 3 0 M
			3 3 0 E
H 0 3 M 13/12		H 0 3 M 13/12	

審査請求 未請求 請求項の数13 F D 外国語出願 (全 67 頁)

(21)出願番号 特願平9-269132

(22)出願日 平成9年(1997) 8月26日

(31)優先権主張番号 9 6 1 0 5 2 0

(32)優先日 1996年8月28日

(33)優先権主張国 フランス (F R)

(71)出願人 591034154

フランス テレコム エス アー

FRANCE TELECOM

フランス国、75015 パリ、プラス・ダ
ル、6

(72)発明者 ラメシュ・ピンディア

フランス・29280・プルサン・アレ・ジャ
ケ・リウ・6

(72)発明者 パトリック・アッド

フランス・29200・プレスト・リュ・ユー
ジェン・ボチエ・4

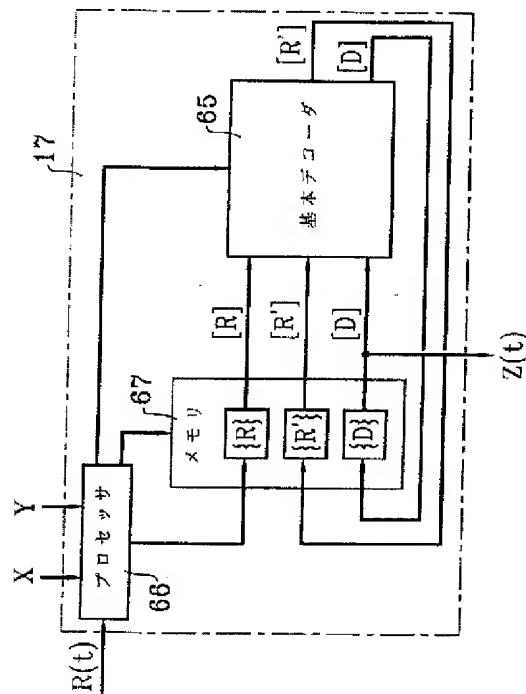
(74)代理人 弁理士 志賀 正武 (外2名)

(54)【発明の名称】 誤り訂正符号を有する情報ビットの変換方法およびこの方法を実行する符号化器と復号化器

(57)【要約】

【課題】 プログラマブルB T C回路の利用を可能にする情報ビットの伝送プロセスを提供すること。

【解決手段】 伝送されるビットが、少なくとも2つの組織ブロック符号の積に従って符号化される。各符号ワード探索ステップ毎に、後続のステップに用いられるデータ行列（{R}）と決定行列（{D}）を決定するため、反復復号化が適用される。各ステップ毎に、入力行列の行または列を復号化することによって、新たな決定行列が決定され、各反復毎に復号化の信頼性を高める訂正項を考慮して、新たなデータ行列が決定される。符号化及び復号化回路（17）は、符号化ブロック当たりの伝送ビット数の選択を可能にするバンクチャリング技法によってプログラム可能になるが、バンクチャリングを施されるビットは、行列の各次元に従って均一に分散された位置を備えることが望ましい。



【特許請求の範囲】

【請求項1】 送信器(10)が、
送信されるべき情報ビット(a_u)から第1の2進行列
($\{a\}$)を形成するステップと、
基本組織ブロック符号(c_1 、 c_2)の積に対応する組織
ブロック符号を適用することによって、第1の2進行列
を第2の2進行列($\{c\}$)に変換するステップと、
第2の2進行列から抽出されたビット(c_v)をチャネ
ルに向けて送信するステップとを実施し、
受信器(15)が、
前記チャネルによって受信された信号($R(t)$)か
ら、第2の2進行列と同じサイズであり、かつ、デジタ
ル・サンプルから構成される入力行列($\{R\}$)を形成
するステップと、
m個の復号化サイクルを具備する入力行列の反復復号化
を行うステップとを実施し、
前記デジタルサンプルの符号がそれぞれ第2の2進行列
におけるビットの初期評価を示し、前記デジタルサンプ
ルの絶対値がそれぞれ前記初期評価に関連した信頼性の
程度を示し、各復号化サイクルは、積符号で用いられる
各基本ブロック符号に関する符号ワード探索ステップ
(32、33)から逐次構成され、
各符号ワード探索ステップ(32、33)毎に、2進数
を成分とするデータ行列($\{R^d\}$)と決定行列
($\{D\}$)とが受信され、前記データ行列と決定行列と
は、入力行列($\{R\}$)とその2進成分が入力行列のサ
ンプルの符号に対応する行列とによって、反復復号化の
ための最初の探索ステップの前に構成され、後続する探
索ステップのために、その2進成分が第2の2進行列の
ビットの新たなそれぞれの評価を表す新たな決定行列
($\{D\}$)が生成され、そのサンプルの絶対値が、それ
ぞれ、前記新たな評価に関連した信頼性の程度を示す新
たなデータ行列($\{R^d\}$)が生成され、
復号化情報ビット◎

【数1】

$$(\hat{a}_u)$$

が、最後の符号ワード探索ステップ中に生成された決定
行列から抽出され、
各符号ワード探索ステップ(32、33)が、受信され
たデータ行列を基本符号の符号ワードにそれぞれ対応す
るデータ・ベクトル($[R^d]$)に分割するステップ
と、この分割に対応して、受信された決定行列を決定ベ
クトル($[D]$)に分割するステップと、少なくとも所
定のデータ・ベクトル/決定ベクトル対に個々に処理を
施すソフト決定(37、42)による基本復号化ステッ
プを含む情報ビットの伝送プロセスにおいて、
第1の2進行列($\{a\}$)は、情報ビット(a_u)に加

えて、受信器に予め知られた値を有する一組のビットを
具備し、前記一組のビットは、第1の2進行列の各次元
に基づいて、ほぼ一定の方法で分散され、前記第1の2
進行列は、組織的な符号化の後、前記第2の2進行列
($\{c\}$)の所定の位置に配置され、前記第2の2進行
列($\{c\}$)は、前記チャネルへ向かっては転送され
ず、受信器(15)は、入力行列($\{R\}$)内の、第2
の2進行列の前記所定位置に対応する位置に、サンプ
ルを配置し、前記サンプルの符号は、前記組のビットの予
め知られた値に個々に対応し、前記サンプルの絶対値
は、最大の信頼性を示すことを特徴とする情報ビットの
伝送プロセス。

【請求項2】 第2の2進行列($\{c\}$)内に他の所定
位置を有する他の組のビットは、前記チャネルへ向かっ
ては転送されず、受信器(15)は、入力行列
($\{R\}$)内の、第2の2進行列の前記他の所定位置に
対応する位置に、サンプルを配置し、前記サンプルの絶
対値は、最小の信頼性を示すことを特徴とする請求項1
記載のプロセス。

【請求項3】 前記他の所定位置は、第2の2進行列
($\{c\}$)の各次元に基づいて、ほとんど一様に分散さ
れることを特徴とする請求項2記載のプロセス。

【請求項4】 データ・ベクトル/決定ベクトル対の処
理を行うソフト決定基本復号化(37、42)が、
データ・ベクトル成分($[R^d]$)の信頼性が最も低い
p個の指標(r_1, \dots, r_p)を決定するステップ
と、

前記p個の指標と決定ベクトル($[D]$)とから復号化
すべきq個の2進ワード($[U^*]$)を構成するステッ
プと、

決定ベクトルと復号化すべきq個の2進ワードとの代数的
復号化に基づいて、q'個の符号ワード($[C^*]$)
を得るステップと、

得られたq'個の符号ワードから、データ・ベクトル
($[R^d]$)に関するユークリッド距離が最も短い符号
ワード($[C^d]$)を選択するステップと、

選択された符号ワード($[C^d]$)の成分とは異なるj
番目の成分を備えたオプションのコンカレント・ワード
($[C^e]$)を決定することによって、かつ、コンカレ
ント・ワードが決定されると、 M^d と M^e とが、それぞ
れ、選択された符号ワード($[C^d]$)とコンカレント
・ワード($[C^e]$)との、データ・ベクトル($[R^d]$)
に関連したユークリッド距離を表し、 C_1^d と R^d
とが、それぞれ、選択された符号ワードとデータ・ベ
クトルとのj番目の成分を表す公式、即ち、◎

【数2】

$$W_j = \left(\frac{|M^e - M^d|}{4} - C_j^d \cdot R_j^d \right) C_j^e$$

を適用することによって、その各成分 W_j が、それぞ

れ、計算される訂正ベクトル（〔W〕）を計算するステップと、

前記選択された符号ワード（〔C^q〕）に等しいとみなされる新たな決定ベクトル（〔D〕）を得るステップと、

訂正ベクトル（〔W〕）に第1の信頼係数（ α_1 ）を乗算した値を、入力行列（〔R〕）から抽出された対応する入力ベクトル（〔R'〕）に加算することによって、新たなデータ・ベクトル（〔R'〕）を計算するステップとを具備することを特徴とする請求項1ないし請求項3のいずれかに記載の伝送プロセス。

【請求項5】 訂正ベクトルの計算ステップにおいて、選択された符号ワード（〔C^q〕）のj番目の成分に関するオプションのコンカレント・ワード（〔C^q〕）の決定に、選択された符号ワード（〔C^q〕）のj番目の成分と、選択された符号ワードを除く、得られたq'の符号ワードの中から、データ・ベクトル（〔R'〕）に関して最短のユークリッド距離を有する候補符号ワードのj番目の成分との比較が含まれることと、前記候補符号ワードのj番目の成分が、選択された符号ワードのj番目の成分と異なる場合には、前記候補符号ワードがコンカレント・ワードとみなされ、等しい場合には、コンカレント・ワードが決まらないことを特徴とする請求項4記載のプロセス。

【請求項6】 訂正ベクトル（〔W〕）の各成分W_jの計算が、該成分に関連したコンカレント・ワードが決定されない場合、 β_1 が第2の信頼係数を表している公式、即ち、◎

【数3】

$$W_j = \beta_1 C_j^q$$

に従って実施されることを特徴とする請求項4または請求項5のいずれかに記載の伝送プロセス。

【請求項7】 訂正ベクトル（〔W〕）の各成分W_jの計算が、該成分に関連したコンカレント・ワードが決定されない場合、 β_1 が第2の信頼係数を表している公式、即ち、◎

【数4】

$$W_j = (\beta_1 - C_j^q R_j') C_j^q$$

に従って実施されることを特徴とする請求項4または請求項5のいずれかに記載の伝送プロセス。

【請求項8】 伝送すべき情報ビット（a_i）から第1の2進行列（〔a〕）を形成する手段（69）と、基本組織ブロック符号（C₁、C₂）の積に相当する組織ブロック符号を適用して、第1の2進行列を第2の2進行列（〔C〕）に変換するように命令されている基本符号化手段（68）と、

第2の2進行列から抽出されたビット（c_v）をチャネ

ルに向けて送信する手段とから構成され、

各第1の行列に含まれるべき情報ビットの数をプログラムするために供給される数Xから、第1の行列の各次元に基づいて一様に分散されるXの位置を決定するために、プログラミング手段（69）が含まれており、第1の行列を生成する手段は、既知の値のビットを前記Xの位置に配置するように設計され、既知の値の前記ビットは、組織的な符号化の後、第2の2進行列の所定位置に配置され、チャネルへ向かっては転送されないことを特徴とする冗長コード（12）。

【請求項9】 プログラミング手段（69）は、チャネルへ向かって転送された第2の行列のビット数をプログラムするために供給される数Yから、第2の行列（〔c〕）内におけるYの位置を決定するように設計されており、前記Yの位置に配置された第2の行列のビットは、チャネルへ向かっては転送されないことを特徴とする請求項8記載のコード。

【請求項10】 プログラミング手段（69）は、第2の2進行列（〔c〕）の各次元に基づいて、ほとんど一定の方法で、前記Yの位置を分散することを特徴とする請求項9記載のコード。

【請求項11】 伝送チャネルによって受信された信号（R（t））から、デジタル・サンプルから構成される入力行列（〔R〕）を形成する手段（66）と、逐次復号化サイクルに従って入力行列を復号化するように命令されている反復復号化手段（65）とから構成され、

前記デジタルサンプルの符号が、基本組織ブロック符号（C₁、C₂）の積に対応する組織ブロック符号を適用する冗長コード（12）によって形成される2進行列のビットのそれぞれの初期評価を示し、前記デジタルサンプルの絶対値が、それぞれ、前記初期評価に関連した信頼性の程度を示し、各復号化サイクルが、積符号で用いられる各基本ブロック符号に関する符号ワード探索ステップ（32、33）から逐次構成され、

各符号ワード探索ステップ（32、33）毎に、復号化手段（65）が、2進数を成分とするデータ行列（〔R'〕）と決定行列（〔D〕）とを受信し、前記データ行列と決定行列とは、入力行列（〔R〕）とその2進成分が入力行列（〔R〕）のサンプルの符号に対応する行列とによって、反復復号化のための最初の探索ステップの前に構成され、後続の探索ステップのために、新たな決定行列（〔D〕）が生成され、その決定行列の2進成分が第2の2進行列のビットの新たなそれぞれの評価を示し、そのサンプルの絶対値が、それぞれ、前記新たな評価に関連した信頼性の程度を示す新たなデータ行列（〔R'〕）が生成され、復号化情報ビット◎

【数5】

5
(\hat{a}_j)

6

が、最後の符号ワード探索ステップ中に生成された決定行列から抽出され、

各符号ワード探索ステップ(32、33)が、受信されたデータ行列を基本符号の符号ワードにそれぞれ対応するデータ・ベクトル($[R^T]$)に分割するステップと、この分割に対応して、受信された決定行列を決定ベクトル($[D]$)に分割するステップと、少なくとも所定のデータ・ベクトル/決定ベクトル対に個々に処理を施すソフト決定(37、42)による基本復号化ステップとを含むエラー訂正デコーダ(17)であって、さらに、決定行列から引き出されるべきデコードされた情報ビットの数をプログラムするために供給される数Xから、前記2進行列の無冗長サブマトリクス($\{a\}$)の各次元に基づいて、一様に分散されたXの位置を決定するために、プログラミング手段(69)を具備し、入力行列($\{R\}$)を形成する手段は、所定符号のデジタルサンプルであり、かつ、その絶対値が最大の信頼性を示すデジタルサンプルを、前記Xの位置に対応する位置に、配置するように設計されていることを特徴とするエラー訂正デコーダ(17)。

【請求項12】 プログラミング手段(66)は、受信された信号から得られる入力行列のサンプル数をプログラムするために供給されるYの数から、入力行列($\{R\}$)内のYの位置を決定するように設計されており、入力行列を形成する手段は、その絶対値が最大の信頼性を示すデジタルサンプルを、前記Yの位置に、配置するように設計されていることを特徴とする請求項11記載のデコーダ。

【請求項13】 プログラミング手段(66)は、入力行列($\{R\}$)の各次元に基づいて、ほとんど一様の方法で、前記Yの位置を分散することを特徴とする請求項12記載のデコーダ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、送信器が、送信すべき情報ビットから第1の2進行列を形成するステップと、基本組織ブロック符号(elementary systematic block codes)の積に相当する組織ブロック符号を適用することによって、第1の2進行列を第2の2進行列に変換するステップと、第2の2進行列から抽出されたビットをチャンネルに向けて送信するステップを実施し、受信器が、前記チャンネルによって受信した信号から、第2の2進行列と同じサイズを備えており、その符号が第2の2進行列におけるビットのそれぞれの初期推定値を表し、その絶対値が、それぞれ、前記初期推定値に関連した信頼性の程度を示す、デジタル・サンプルから構成される入力行列を形成するステップと、それぞれ、積符号

に用いられる各基本ブロック符号に関する符号ワード探索ステップから逐次構成される、いくつかの(mの)復号化サイクルを含む入力行列の反復復号化を行うステップを実施し、各符号ワード探索ステップ毎に、反復復号化のための最初の探索ステップの前に、それぞれ、入力行列と、その2進成分が入力行列のサンプルの符号に対応する行列によって構成される、2進成分によるデータ行列及び決定行列が受信され、後続の探索ステップのために、その2進成分が第2の2進行列のビットの新たなそれぞれの推定値を表す、新たな決定行列が生成され、そのサンプルの絶対値が、それぞれ、前記新たな推定値に関連した信頼性の程度を示す、新たなデータ行列が生成され、復号化情報ビットが、最後の符号ワード探索ステップ中に生成された決定行列から抽出され、各符号ワード探索ステップに、受信したデータ行列を、それぞれ、基本符号の符号ワードに対応するデータ・ベクトルに分割するステップと、相応じて、受信した決定行列を決定ベクトルに分割するステップと、それぞれ、少なくとも所定のデータ・ベクトル/決定ベクトル対に処理を施すソフト決定による基本復号化のステップが含まれる情報ビットの伝送プロセスに関するものである。

【0002】

【従来の技術および発明が解決しようとする課題】情報(音声、イメージ、データ等)のデジタル伝送の問題に関して、情報源符号化とチャンネル符号化とは、通常、弁別される。情報源符号化は、伝送されるべき信号の2進表現を形成する。それは、通常、伝送されるべき信号の内容の関数として設計される。近年になって、良好な伝送の質を保持したまま、デジタル・レートを低下させるために、情報源符号化の問題に対して多大の努力が費やされるようになってきている。しかし、これらの新たな情報源符号化技法は、伝送中の摂動からビットをより有効に保護することが必要になる。さらに、高周波成分(ノイズ要素、電力飽和)の物理的及び経済的制限、並びに、伝送に許容される電力レベルに対する規制によって、デジタル伝送システムのレンジが制限されることになる。

【0003】このため、チャンネル符号化の問題、とりわけ、ブロック符号化の問題に関して、多大の労力が払われてきた。このタイプのエラー訂正符号化は、情報源符号化によって生じるkの情報ビットにn-kの冗長ビットを加え、いくつかの伝送エラーを訂正するため、受信時にこれらの冗長ビットを用いることにある。比 $R = k/n$ は、符号の効率と呼ばれる。符号化利得Gは、与えられた2進エラー・レート(BER)を実現するために、符号化を伴わない場合と伴う場合とにおける、受信器の入力で必要な情報ビット当たりのエネルギーEb間における比(デシベルで表示される)と定義される。一般的な目的は、(i)符号化利得Gが、できるだけ高くなり(BER = 10^{-5} の場合、 $G > 5$)、(ii)符号効

率 R が、できるだけ高くなり($R > 0.6$)、 (i, i) 復号化の複雑さが、できるだけ低くなるように、コーダ、及び、特にそれに関連したデコーダを創り出すことにある。

【0004】デジタル情報を記憶するケースは、伝送の特殊ケースとみなされる。そのケースでは、伝搬チャネルは、情報がより長いあるいはより短い時間にわたって記憶された状態に保たれるメモリを含み、送信器及び受信器は、同じであっても、同じでなくてもかまわない。従って、一般的に、チャネル符号化及び関連する復号化

の概念が、伝送と同じ方法で、情報の記憶分野にも適用可能であることは理解される。訂正すべきエラーは、メモリにおける読み書き、あるいは、メモリの内容の変更、あるいは、メモリにおける読み書き装置との通信(遠隔であろうとなかろうと)に起因するものである。【0005】連結技法を利用することによって、エラー訂正符号の性能が向上することが、知られている。特に、本発明にとりわけさらに関連性の深い積符号技法は、2つの単純なブロック符号(即ち、小さな最小ハミング距離 d を備えたブロック符号)から、その最小ハミング距離が、用いられている基本符号のハミング距離との積に等しい符号を得ることを可能にする(1970年9月の、IEEE Trans. on Information theory, Vol. IT-16, No. 5, 624~627ページにおける、S. M. Reddyによる「On decoding iterated codes」を参照されたい)。

【0006】もし、パラメータ (n_1, k_1, d_1) を備えたブロック符号が、 C_1 によって指定され、パラメータ (n_2, k_2, d_2) を備えたブロック符号が、 C_2 によって指定されるならば、 C_1 と C_2 との積である符号の適用は、行列内における $k_1 \times k_2$ の連続した情報ビットの順序づけ、及び、符号 C_2 による行列の k_1 行の符号化、さらに、符号 C_1 による結果の行列の n_2 列の符号化において行われる。積符号 P のパラメータは、従って、 $(n = n_1 \times n_2; k = k_1 \times k_2; d = d_1 \times d_2)$ によって得られる。符号 P の効率 R は、 $R_1 \times R_2$ に等しい。事後最大尤度(MLP)に基づく符号 P の復号化は、最適性能を実現することを可能にする。従って、最大漸近的符号化利得は、関係式 $G < 10 \log_{10} (R \cdot d)$ によって

近似されることができる。【0007】積符号は、従って、極めて有益であるが、MLPによる復号化は、短いブロック符号の場合を除けば、一般に、あまりにも複雑である。

【0008】1993年5月のProc. ICC' 93, Geneva, 1740~1745ページの「Separable MAP filtes for the decoding of product and concatenated codes」と題する論文において、J. Lodge他は、データ行から抽出され

た行ベクトル及び列ベクトルが、ビットに関する対数尤度比(LLR)を推定するBah1のアルゴリズム(1974年3月のIEEE Trans. on Information Theory, Vol. IT-20, 248~287ページにおけるL. R. Bah1他による「Optimal decoding of linear codes for minimizing symbol error rates」を参照されたい)を利用して復号化される、序文において簡単に述べたタイプの反復復号化アルゴリズムを提案している。Bah1のアルゴリズムによれば、LLRによって表現されるソフト決定が得られ、MLPに近い性能を実現することが可能になる。しかし、いくつかの状態が $n-k$ の関数として指数関数的に増大する復号化トレリスが利用される。結果として、Lodge他

のアルゴリズムは、例えば、ハミングの符号(16, 11, 3)のような短い符号には適しているが、例えば、BCH符号(63, 51, 5)のような効率の高い符号に実施する場合には役に立たないことが立証されている。

【0009】始めに簡単に述べたタイプのもう1つのプロセスが、欧州特許出願第0 654 910号に提示されており、その内容については、本解説に組み込まれている。【0010】後者のプロセスは、線形ブロックをなす符号から構成され、代数デコーダが利用可能な、全ての積符号の復号化に利用することが可能である。このプロセスによって得られる性能は、ほぼ最適である(Proc. IEEE GLOBECOM' 94 Conference, Vol. 1/3, Nov. - Dec. 1994, San Francisco, 339~343ページのR. Pyndiah他による「Near optimum decoding of product codes」を参照されたい)。実際のところ、それを利用すると、問題となる積符号に関する理論上のシャノン限界を超える2.5 dBの領域における S/N 比に関して、 10^{-5} に等しいBERを得ることができる(所定の積符号に関して、4回反復して)。さらに、このプロセスは、Lodge他の提案した解決法に比べるとはるかに単純で、はるかに信頼性が高い。従って、基本符号長 n_1, n_2 が256までの極めて大きい積符号を復号化することが可能である。

【0011】今後、ブロック・ターボ符号回路またはBTC回路(1995年9月のProc. GRETSI' 95 Conference, Vol. 2, 981~984ページのO. Raoul他による「Architecture and design of a turbo decoder circuit for product codes」を参照されたい)と呼ばれる、序文において述べたタイプの反復復号化を実施するのに適した回路の創出を考慮すると、該回路の表面積は、いく

つかの基本デコーダを連結するのではなく、同じ基本デコーダを利用して、いくつかの反復を実施することによってかなり縮小することができるのは明らかである。企図されている用途に応じて、反復数は、基本デコーダの複雑さの関数になる。基本デコーダの複雑さが低下するほど、反復数が増加し、従って、基本デコーダの複雑さを低下させる利点が増大する。

【0012】経済的理由から、異なる数の冗長ビットを備えた、多様なサイズのデータ・ブロックの処理を可能にするプログラマブルBTC回路を備えることが望ましい。従って、さまざまな用途に同じBTC回路の利用を企図することが可能であり、これにより、開発コストに関してかなりの節約が可能になる。

【0013】本発明の主たる目的は、この必要に応えて、プログラマブルBTC回路の利用を可能にする情報ビットの伝送プロセスを提供することにある。

【0014】

【課題を解決するための手段】従って、本発明は、序文において述べたタイプのプロセスにおいて、第1の2進行列は、情報ビットに加えて、受信器に予め知られた値を有する一組のビットを含み、前記一組のビットは、第1の2進行列の各次元に基づいて、ほとんど一様の方法で分散され、前記第1の2進行列は、組織的な符号化の後、前記第2の2進行列内の所定位置に配置され、前記第2の2進行列は、チャンネルへ向かっては転送されず、受信器は、入力行列内の、第2の2進行列の前記所定の位置に対応する位置に、サンプルを配置し、前記サンプルの符号は前記組のビットの予め知られた値に個々に対応し、前記サンプルの絶対値は最大の信頼性を示す、ということを提案する。

【0015】本発明は、単純ブロック符号の分野でよく知られている短縮化技術に類似した技術を利用する。 n 、 k 、及び、 d を下記の形式の積符号パラメータとする。◎

【数6】

$$n = \prod_{i=1}^L n_i, \quad k = \prod_{i=1}^L k_i, \quad \text{and} \quad d = \prod_{i=1}^L d_i$$

ここで、 L は、それぞれのパラメータが (n_i, k_i, d_i) である基本符号の数である（一般性を制限することなく、 $L=2$ のケースについて、以下で考察される）。 k 及び n は、「第1」と「第2」の2進行列におけるそれぞれのビット数である。

【0016】本発明は、行列に含まれる独立情報ビットの数を、 k 以下の任意の数 $k-X$ に適応させることが可能であり、受信器の復号化回路は、既知のビット数 X がいくらであろうと同じである。これらの X ビットの位置は、第1の行列内に一様に分散し、このことは、反復復号化プロセスの性能を最適に利用できるようにする。こ

の点において、1つあるいは2つ以上の符号を基本ブロックに短縮化することは、（ X の値の選択の幅が狭くなり、とりわけ、BER利得が生じないある種の基本符号化につながるというように）利点が少ない、という点は注意されるべきである。

【0017】バンクチャリングを施される積符号のパラメータ (n', k', d') は、最終的に、 $n' = n - Y$ 、 $k' = k$ 、及び、 $d' = d$ になる。その効率 R' は、 $R' = k / (n - Y) \geq k / n$ になる。

10 【0018】有利な実施形態の場合、第2の2進行列内の所定位置の他の組の Y ビットは、チャンネルへ向かっては転送されない。受信器は、入力行列内において、第2の2進行列のこれら Y の所定位置に対応する位置に、その絶対値が最小の信頼性を表すサンプルを配置する。

【0019】ここでは、重畳符号の分野でしばしば使用されるバンクチャリング技術に類似したバンクチャリング技術が使用される。バンクチャリングは、符号の効果を増大する。重畳符号の場合、その目的は、一般的に、2進符号を用いる間に、 $1/2$ より多い符号化効率を達成することである。即ち、この符号化効率は最も単純な復号化配列格子を有する。一般的に、破壊された重畳符号は、同一効果の破壊されていない符号の距離特性に類似した距離特性を有する。

【0020】しかしながら、バンクチャリングは、通常、ブロック符号には、適用されない。実際、最適の距離特性を持った高い効率を有するブロック符号が多く存在する。故に、バンクチャリングが複雑性の中の利得を達成することなく距離特性を減じる、ということは予測される。上記複雑性は、重畳符号の場合と同様に、容易に判断できる。発明者は、積符号の場合において、上記のように適用されたバンクチャリング（即ち、ソフト決定反復符号化処理と結合されたバンクチャリング）が、符号器の性能を著しくは減じなかった、ということを観察したことに驚いた。

30 【0021】そして、 Y パラメータは、符号器の出力ブロック当たりのビット数と、全体的な符号化効率とをプログラムするのに使用されることができる。例えば、（数 X を介した）ブロック当たりにおける冗長でない情報ビット数のプログラミングが、符号化効率における過度の低下につながる時、もし、 $Y > 0$ ならば、より高い効率が達成される。短縮化され、かつ、バンクチャリングを施された積符号のパラメータ (n'', k'', d'') は、最終的に、 $n'' = n - X - Y$ 、 $k'' = k - X$ 、及び、 $d'' = d$ である。その効率 R'' は、 $R'' = (k - X) / (n - X - Y)$ である。

50 【0022】データ・ベクトル／決定ベクトル対を処理するためのソフト決定基本復号化は、特に、EP-A-0 654 910に記載のタイプとすることが可能である。従って、それは、データ・ベクトル成分の信頼性が最も低いいくつかの（ p の）指標を決定するステップ

と、前記pの指標及び決定ベクトルから復号化すべきいくつかの(qの)2進ワードを構成するステップと、決定ベクトル、及び、復号化すべきqの2進ワードの代数的復号化に基づいて、q'の符号ワードを得るステップと、得られたq'の符号ワードから、データ・ベクトルに関するユークリッド距離が最も短い符号ワードを選択するステップと、その各成分 W_i^d が、選択された符号ワードとは異なるj番目の成分を備えたオプションのコンカレント・ワードを決定し、コンカレント・ワードが決定されると、 M^d 及び M^e が、それぞれ、選択された符号ワード及びコンカレント・ワードの、データ・ベクトルに関連したユークリッド距離を表し、 C_i^d 及び R_i^e が、それぞれ、選択された符号ワード及びデータ・ベクトルのj番目の成分を表すことになる公式、即ち、⑦

$$W_j = \left(\frac{|M^e \cdot M^d|}{4} \cdot C_j^d \cdot R_j^e \right) C_j^d$$

を適用することによって、それぞれ、計算される、訂正ベクトルを計算するステップと、前記選択された符号ワードに等しいとみなされる新たな決定ベクトルを得るステップと、訂正ベクトルに第1の信頼係数を掛けた値を、入力行列から抽出した対応する入力ベクトルに加算するステップから構成される。

【0023】望ましい実施形態の場合、訂正ベクトルの計算ステップにおいて、選択された符号ワードのj番目の成分に関するオプションのコンカレント・ワードの決定には、選択された符号ワードのj番目の成分と、選択された符号ワードを除く、得られたq'の符号ワードの中から、データ・ベクトルに関して最短のユークリッド距離を有する候補符号ワードのj番目の成分との比較が含まれており、前記候補符号ワードのj番目の成分が、選択された符号ワードのj番目の成分と異なる場合には、前記候補符号ワードがコンカレント・ワードとみなされ、異なる場合には、コンカレント・ワードが決まらない。

【0024】本発明の第2の態様は、伝送すべき情報ビットから第1の2進行列を形成するための手段と、基本組織ブロック符号の積に相当する組織ブロック符号を用いて、第1の2進行列を第2の2進行列に変換するように命令されている基本符号化手段と、第2の2進行列から抽出されたビットをチャンネルに向けて送信するための手段から構成され、各第1の行列に含まれるべき情報ビットの数をプログラムするために供給される数Xから、第1の行列の各次元に基づいて一様に分散されたXの位置を決定するために、プログラミング手段を具備し、第1の行列を生成する手段は、既知の値のビットを前記Xの位置に配置するように設計され、既知の値の前記ビットは、組織的な符号化の後、第2の2進行列の所定位置に配置され、チャンネルへ向かっては転送されないプログ

ラムブル冗長コードに関するものである。

【0025】本発明の第3の態様は、伝送チャンネルによって受信した信号から、その符号が、基本組織ブロック符号の積に相当する組織ブロック符号を用いて、冗長コードによって形成される2進行列のビットのそれぞれの初期推定値を表し、その絶対値が、それぞれ、前記初期推定値に関連した信頼性の程度を示す、デジタル・サンプルから構成される入力行列を形成するための手段と、それぞれ、積符号に用いられる各基本ブロック符号に関する符号ワード探索ステップから逐次構成される、逐次復号化サイクルに従って入力行列の復号化を行うように命令されている反復復号化手段から構成され、各符号ワード探索ステップ毎に、復号化手段が、最初の探索ステップの前に、それぞれ、入力行列と、その2進成分が入力行列のサンプルの符号に対応する行列によって構成される、2進成分によるデータ行列及び決定行列を受信し、後続の探索ステップのために、その2進成分が第2の2進行列のビットの新たなそれぞれの推定値を表す、新たな決定行列が生成され、そのサンプルの絶対値が、それぞれ、前記新たな推定値に関連した信頼性の程度を示す、新たなデータ行列が生成され、復号化情報ビットが、最後の符号ワード探索ステップ中に生成された決定行列から抽出されるようになっており、各符号ワード探索ステップに、受信したデータ行列を、それぞれ、基本符号の符号ワードに対応するデータ・ベクトルに分割するステップと、相応じて、受信した決定行列を決定ベクトルに分割するステップと、それぞれ、少なくとも所定のデータ・ベクトル/決定ベクトル対に処理を施すソフト決定による基本復号化のステップが含まれており、さらに、決定行列から引き出されるべき復号化された情報ビットの数をプログラムするために供給される数Xから、前記2進行列の無冗長サブマトリクスの各次元に基づいて、一様に分散されたXの位置を決定するために、プログラミング手段を具備し、入力行列を形成する手段は、所定の符号のデジタルサンプルであり、かつ、その絶対値が最大の信頼性を示すデジタルサンプルを、前記Xの位置に対応する位置に、配置するように設計されているプログラマブル・エラー訂正デコーダに関するものである。

【0026】本発明の他の特徴及び利点については、付属の図面と併せて読むべき、非制限的实施形態に関する以下の説明から明らかになるであろう。

【0027】

【発明の実施の形態】発明者は、EP-A-0 654 910に記載の積符号に関する反復復号化プロセスの、良好な性能/複雑さの妥協点をもたらす、有利な改変案を開発した。以下では、本発明による積符号の構造に関するより詳細な説明の前に、図1～5に関連してこの改変案の説明を行うことにする。一方では、前記改変案は、任意のタイプの積符号の反復復号化に適用可能で

10

20

30

40

50

あり、他方では、本発明による伝送プロセスは、例えば、EP-A-0 654 910や、あるいは、前述のJ. Lodge他による論文にも記載されているような他の反復符号化方法に適合することが可能であることが分かる。

【0028】図1に示す伝送連鎖の場合、伝送すべき情報ビット a_j が、送信器10のチャネル・コードに対する入力にアドレス指定された信号 $X(t)$ に含まれる。

◎

【数8】

$$X(t) = \sum_j a_j h(t - jT)$$

この信号 $X(t)$ は、アナログ信号 $S(t)$ から情報源コード11によって形成される。情報源コード11によって、普通、 a_j sは、独立したビットとなり、等しい確率で0か1の値をとることになる。 $h(t)$ は、2つの連続したビットを分離する時間間隔である継続時間ゲート T を表している。チャネル・コード12は、ブロック符号化を用いて、信号 $Y(t)$ を発生する。◎

【数9】

$$Y(t) = \sum_j c_j h(t - jT)$$

ここで、 c_j は符号化ビットであり、 T' は、2つの符号化ビットを分離する時間間隔である($T' < T$)。変調器13は、シーケンス $Y(t)$ を伝搬チャネルに適合する信号シーケンスに変換する。無線チャネルに関連した2状態移相打鍵の場合、送り出される信号例は、下記

によって示される。◎

【数10】

$$E(t) = \sum_j e_j h(t - jT_s) \sin(2\pi f_0 t)$$

ここで、 f_0 は、搬送波の周波数であり、 $e_j = 2, c_j - 1$ である。受信器15のアンテナで受信された信号は、係数 α だけ減衰する。復調器16は、各ビット毎に、下記のように表すことが可能な確率比を導き出す。

$R_j = e_j + B_j$

ここで、サンプル B_j は、伝搬チャネルによって導入されるノイズ・サンプルであり、ビット c_j とは無関係であり、互いに相関せず、平均が0で、標準偏差 σ は、 S/N 比によって決まる。従って、復調器16の出力における信号は、下記に等しい。◎

【数11】

$$R(t) = \sum_j R_j h(t - jT)$$

【0029】チャネル・デコーダ17は、次に、エラー

(8)

特開平10-135849

を最小限に抑えるため、送信に用いられるチャネル符号化を利用して、送信されるビットに関連した決定を行う。その出力は、下記によって示される。◎

【数12】

$$Z(t) = \sum_j \hat{a}_j h(t - jT)$$

ここで、◎

【数13】

10

\hat{a}_j

は、チャネル・デコーダによって行われた決定である。情報源デコーダ18は、次に、チャネル・デコーダ17によって供給されるビットからアナログ信号 $S(t)$ を再構成する。

【0030】本発明は、主として、チャネル・コード12及びチャネル・デコーダ17に属する。従って、云うまでもなく、さまざまなタイプの情報源符号化／復号化、変調／復調、及び、伝搬チャネルに適合する。とりわけ、本発明は、デジタル・テレビジョンに関連して適用することが可能である。コード11及びデコーダ18は、例えば、MPEG(動画像圧縮方式の標準化作業グループ)規格に基づいて作製することが可能であり、変調器13及び復調器16は、用いられている伝搬チャネル(無線、電線等)に適合する。もう1つの応用例は、ファクシミリ伝送である。

【0031】チャネル12によって用いられるブロック符号は、基本組織符号から得られる積符号である。後述の実施形態の場合、それは、それぞれ、パラメータ(n_1, k_1, d_1)及び(n_2, k_2, d_2)を備えた2つの線形ブロック符号 C_1, C_2 の積である。

【0032】図2には、従来の符号化回路で実施される符号化手順が示されている。情報源コード11から逐次受信するビット a_j は、まず第1に、 k_1 行及び k_2 列からなる行列 $\{a\}$ に従って、 $k_1 \times k_2$ ビットのグループによって保持される(ステップ21)。次に、ブロック符号 C_2 が、行列 $\{a\}$ の k_1 行に適用され、これによって、 k_1 行と n_2 列からなる行列 $\{b\}$ が得られる(ステップ22)。符号 C_2 は、組織的であるので、行列 $\{b\}$ の n_2 列の k_2 は、行列 $\{a\}$ の、例えば、最初の k_2 列と同じである。次に(ステップ23)、ブロック符号 C_1 が、行列 $\{b\}$ の n_2 列のそれぞれに適用され、これによって、 n_1 行と n_2 列からなる行列 $\{c\}$ が得られるが、その c_j 成分は、信号 $Y(t)$ の形で変調器13に逐次伝送されるビットである(ステップ24)。符号 C_1 は、組織的であるので、行列 $\{c\}$ の n_1 行の k_1 は、行列 $\{b\}$ の、例えば、最初の k_1 行と同じである。従って、行列 $\{c\}$ の k_1 行と k_2 列の上方左側部分は、行列 $\{a\}$ と同じであり、行列 $\{c\}$ の他の成分は、冗長ビットである。行列 $\{c\}$ の全ての列が、符号 C_1 の符号ワードである。同様に、基本符号が線形であ

50

れば、行列{c}の全ての行が、符号C₂の符号ワードである。

【0033】チャンネル・デコーダ17は、反復復号化手順を適用するが、その概要フローチャートが、図3に示されている。チャンネル・コーダ12によって形成された符号化ブロックの送信に相応じた、復調器16から受信する信号R(t)のn₁×n₂のサンプルR_{j₁,j₂}(1≤j₁≤n₁, 1≤j₂≤n₂)からなるブロックの受信後、これらのサンプルは、n₁行及びn₂列からなる入力行列{R}に保持される(ステップ39)。

【0034】このn₁×n₂のサンプルからなるブロックの復号化は、計数変数iを0に初期設定し、その成分が当初入力行列{R}の成分と同じである、n₁行及びn₂列からなるデータ行列{R'}を形成し、その成分が、2進数(-1または+1)であり、当初、それぞれ、入力行列{R}の対応する成分の符号、即ち、D_{j₁,j₂}=sgn(R_{j₁,j₂})=±1を表す、n₁行及びn₂列からなる決定行列{D}を形成することによって初期化される(ステップ31)。

【0035】この初期化後、反復復号化には、いくつかの(mの)復号化サイクルが含まれている。各復号化サイクルには、逐次、データ行列の列内において符号C₁のワードを探索するステップ32、データ行列の行内において符号C₂のワードを探索するステップ33が含まれる。

【0036】各探索ステップ32または33において、決定行列{D}及びデータ行列{R'}の成分の新たな値が計算され、後続の探索ステップに用いられる。各探索ステップ32または33は、この行列の成分R'_{j₁,j₂}におけるノイズ・サンプルB_{j₁,j₂}の生起を低減するため、データ行列{R'}に施されるフィルタリングとみなすことが可能である。

【0037】ステップ32及び33は、行列の行と列の役割を交換すると、ほぼ同じになる。探索ステップ32の初期化時には、計数変数が、1単位だけインクリメントされ、列指標j₂は、1に初期設定される。行列{R'}のj₂番目の列に対応するデータ・ワードの復号化

$$\hat{a}_{j_1,j_2} = D_{j_1,j_2} \quad (1 \leq j_1 \leq k_1, 1 \leq j_2 \leq k_2)$$

である。これらの◎

【数17】

$$\hat{a}_{j_1,j_2}$$

は、値が-1または+1であり、値0または1をとるように簡単に変換することが可能である。

【0038】本発明の最初の実施形態における、データ行列の列に対応するデータ・ワードの復号化のステップ37が、図4のフローチャートに詳細に示されている。このステップ37において、長さn₁のデータ・ベクトル[R']及び決定ベクトル[D]に処理を施して、そ

*が、符号C₁に基づいて実施され(ステップ37)、これによって、行列{D}及び{R'}の成分D_{j₁,j₂}及びR'_{j₁,j₂}の新たな値が得られる(1≤j≤n₁)。復号化ステップ37に後続して、列指標j₂と列n₂の数との比較38が行われる。j₂がn₂未満のままであれば、指標j₂が1単位だけインクリメントされ(ステップ39)、復号化ステップ37が繰り返される。j₂がn₂に等しくなると、全て列の処理が済んだことになり、進行中の復号化サイクルの他の符号ワード探索ステップ33が開始される。探索ステップ33の初期化時には、計数変数が、1単位だけインクリメントされ、行指標j₁は、1に初期設定される。行列{R'}のj₁番目の列に対応するデータ・ワードの復号化が、符号C₂に基づいて実施され(ステップ42)、これによって、行列{D}及び{R'}の成分D_{j₁,j₂}及びR'_{j₁,j₂}の新たな値が得られる。復号化ステップ42に後続して、符号C₁の行指標j₁とパラメータn₁の比較43が行われる。j₁がn₁未満のままであれば、指標j₁が1単位だけインクリメントされ(ステップ44)、復号化ステップ42が繰り返される。j₁がn₁に等しくなると、符号ワード探索ステップ33が終了し、計数変数iが2mと比較される(テスト45)。iが2m未満のままであれば、探索ステップ32に再入し、後続の復号化サイクルを開始する。iが2mに等しくなると、mの復号化サイクルが実施されたことになり、k₁×k₂の復号化情報ビット◎

【数14】

$$\hat{a}_{j_1,j_2}$$

が、最後の符号ワード探索ステップ33中に生成された決定行列{D}から抽出される。図2に関連して上述のやり方で組織符号C₁、C₂を適用すると、◎

【数15】

$$\hat{a}_{j_1,j_2}$$

は、行列{D}の最初のk₁行と第1のk₂列において簡単に回復することができる。即ち、◎

【数16】

40 それぞれ、データ行列{R'}及び決定行列{D}の細分化部、即ち、R'_{j₁}=R'_{j₁,j₂}及びD_{j₁}=D_{j₁,j₂}(1≤j≤n₁)が構成される。最初に(ステップ51)、ベクトル[R']の最も信頼性の低いpの成分、即ち、2進決定しきい値(ゼロ)に最も近い[R']の成分にマークが付けられる。これら最も信頼性の低いpの成分に対応する指標が、r₁、r₂、...、r_pで表され、一例として、下記のようなになる。◎

【数18】

$$|R'_{r1}| < |R'_{rj}| \quad \forall j \neq r1$$

$$|R'_{r2}| < |R'_{rj}| \quad \forall j \neq r1, r2$$

【0039】これらpの指標が識別されると、長さ n_1 のqの2進テスト・シーケンスが構成され、さらに、qのテスト・シーケンスのそれぞれと決定ベクトル[D]を組み合わせ、長さ n_1 の復号化すべきqの2進ワード $[U^1], \dots, [U^q]$ が構成される(ステップ52)。各ワード $[U^s]$ は、pの指標 $r1, \dots, rp$ に対応する成分以外のその全ての成分が決定ベクトル[D]の対応する成分に等しくなるように構成される。*

$$\begin{aligned} * \quad [T^7] &= [T^1] \oplus [T^2] \\ [T^8] &= [T^1] \oplus [T^3] \\ [T^9] &= [T^1] \oplus [T^4] \\ [T^{10}] &= [T^1] \oplus [T^5] \\ [T^{11}] &= [T^1] \oplus [T^6] \\ [T^{12}] &= [T^2] \oplus [T^3] \\ [T^{13}] &= [T^2] \oplus [T^4] \\ [T^{14}] &= [T^2] \oplus [T^5] \end{aligned}$$

ここで、 r は、2つのベクトル間における成分毎の排他的OR演算を表す。◎

【数20】

$$* [U^s] = [T^s] \oplus [D] \quad \text{for } 1 \leq s \leq q \quad 30$$

【0040】後続ステップ53において、決定ベクトル[D]及びqのワード $[U^s]$ の代数的復号化が実施される。この代数的復号化に関して、BCH符号の場合には、例えば、ブロック符号化の分野において周知のBerlekampデコーダが用いられる(1968年、ニューヨークのマグローヒル社から出版されたE. R. Berlekampによる「Algebraic Coding Theory」を参照されたい)。q+1の基本復号化によって、符号 C_1 の q' の符号ワード

$[C^1], \dots, [C^q]$ が得られる。一般的な場合、一方では、所定の符号ワードが、復号化結果に数回にわたって現れることもあり、他方では、信号がかなり歪んでいると、代数デコーダが所定の符号ワードを見つけないこともあるので、 $q' \leq q+1$ になる。従って、代数的復号化の結果として得られるワードをチェックして、符号 C_1 のワードを構成するか否かを判定しなければならない。このチェックは、符号 C_1 に関するパリティ検査行列によって得られた各ワードの乗算を行い、乗算結果がゼロでない場合、そのワードを消去す

* $U_i^s = D_i$ ($j \neq r1, \dots, rp$ の場合)

それは、一般に、ベクトル[D]の対応する成分とは異なる1つまたは2つの成分だけしか備えないワード $[U^s]$ を考慮すると、十分である。q=p(p+1)/2の場合、これら全てのワードが考慮される。例えば、p=6で、q=21の場合、シーケンス $[T^s]$ 及び $[U^s]$ ($1 \leq s \leq q$)は、下記のようにして構成することが可能である。

* p=6の最初のテスト・シーケンス $[T^s]$ は、位置 rs に+1に等しいビットを備え、他の位置に-1に等しいビットを備える。即ち、 $T_{rs}^s = +1$ 及び $T_{ij}^s = -1$ ($1 \leq s \leq 6$ 及び $j \neq rs$ の場合)である。◎
【数19】

$$\begin{aligned} [T^{15}] &= [T^2] \oplus [T^6] \\ [T^{16}] &= [T^3] \oplus [T^4] \\ [T^{17}] &= [T^3] \oplus [T^5] \\ [T^{18}] &= [T^3] \oplus [T^6] \\ [T^{19}] &= [T^4] \oplus [T^5] \\ [T^{20}] &= [T^4] \oplus [T^6] \\ [T^{21}] &= [T^5] \oplus [T^6] \end{aligned}$$

ることによって、簡単に実施することが可能である。しかし、符号 C_1 が完全な場合(即ち、とりわけ、ハミング符号の場合に当てはまることであるが、 n_1 ビットのどのワードも、可能性のある全ての符号ワードからの間隔が $(d_1-1)/2$ を超えない場合)、代数デコーダからの結果をチェックするステップは、無駄である。

【0041】見つかった q' の符号ワードの中から、データ・ベクトル $[R']$ に関して最短のユークリッド距離 $M^d = \|[C^d] - [R']\|^2$ を示すもの $[C^d]$ が選択される(ステップ54)。このワード $[C^d]$ は、次の決定ベクトルを構成する。さらに、ワード $[C^d]$ を除くqの符号ワードの中から、候補符号ワード $[C^e]$ として、データ・ベクトル $[R']$ からのユークリッド距離 $M^e = \|[C^e] - [R']\|^2$ が最短のものが選択される。この候補ワードは、ワード $[C^d]$ の個々のビットに関連した信頼度の計算のためのコンカレント・ワードとして利用することが可能な唯一のワードになる。

【0042】次に、訂正ベクトル $[W]$ ($1 \leq j \leq n_1$)の成分 W_j からの計算のためにループが実施される。このループの開始時に(ステップ55)、成分指標jが1に初期設定される。このループの各反復毎に、テスト・ステップ56を実施して、候補ワード $[C^e]$ のj番目の成分が、選択された符号ワード $[C^d]$ のj番目の成分と異なるか否かの判定が行われる($C_j^e \neq$

C_i^d)。異なる場合、候補符号ワード $[C^e]$ は、 j 番目の成分に関するコンカレント・ワードになる。次に、ステップ58で、下記の公式に従って成分 W_j の計算が行われる。◎

【数21】

$$W_j = \left(\frac{M^c - M^d}{4} - C_j^d \cdot R_j' \right) \cdot C_j^d$$

【0043】この公式に生じる量 $M^c - M^d$ は、常に正であり、従って、 $|M^c - M^d| = M^c - M^d$ になることが分かる。テスト・ステップ56によって、 $C_i^c = C_i^d$ になることが明らかになると、即ち、コンカレント・ワードを決定することができなければ、ステップ59において、下記公式に基づいて成分 W_j の計算が行われる。◎

【数22】

$$W_j = (\beta_i - C_j^d \cdot R_j') \cdot C_j^d$$

ここで、 β_i は、正の信頼係数を表している。訂正成分 W_j の計算後、成分指標 j がベクトル $[R']$ の長さ n_1 と比較される(ステップ60)。 j が n_1 未満のままであれば、指標 j は1単位だけインクリメントされ(ステップ61)、後続の反復プロセスが、テスト56から始めて、実施される。

【0044】 j が n_1 に等しくなると、ループが終了し、復号化ステップ37が、データ・ベクトル $[R']$ 及び決定ベクトル $[D]$ の更新62によって終了する。新たなベクトル $[R']$ は、入力ベクトル $[R]$ (その各成分 R_i が入力行列 $\{R\}$ から抽出される、即ち、 $R_i = R_{1,i+1}$) と、訂正ベクトル $[W]$ に別の信頼係数 α_i を掛けた値との和に等しいとみなされる。即ち、 $[R'] = [R] + \alpha_i [W]$ である。新たな決定ベクトル $[D]$ は、ステップ54において選択された符号ワード $[C^d]$ に等しいとみなされる。

【0045】改変実行案の場合、ステップ59において適合する場合に用いられる公式(2)が、次のように置き換えられる。◎

【数23】

$$W_j = \beta_i C_j^d$$

これによって、新たな決定 C_i^d の符号に正比例する訂正 W_j が得られる。コンカレント・ワードが識別されなければ、信頼係数を利用する他の公式を利用することも可能である。

【0046】データ行列の行に対応するデータ・ワードの復号化ステップ42は、符号 C_1 を符号 C_2 に置き換え、長さ n_1 を長さ n_2 に置き換え、行列 $\{R'\}$ 、 $\{D\}$ 、 $\{R\}$ を列ベクトル $[R']$ 、 $[D]$ 、 $[R]$ に分割せず、行ベクトルに分割すれば、図4に関連して上で詳述のステップ37と同様である。

【0047】信頼係数 α_i 及び β_i は、図3のフローチャートに示す計数変数 i に対応する指標が割り当てられる。実際、該係数 α_i 及び β_i は、探索ステップ32、33の一方ともう一方の間で変化する可能性がある。 α_i 及び β_i の値は、符号ワード探索ステップ32、33の進行につれて増大し、復号化の信頼性の向上を反映するのが望ましい。

【0048】上記復号化プロセスの性能の例示として、図5には、2つの同じ基本符号 BCH (64、57、4) の積の場合に、シミュレーションによって得られる S/N 比 E_B/n_0 の関数としての曲線 BER が示されている。この場合、行及び列の復号化(ステップ37、42)に用いられる基本デコーダは、復号化が EP-A-0 694 910 に解説の最適バージョンに従う場合、約43000の論理ゲートを必要とする。これら43000のゲートの中から、25000のゲート、即ち、回路の59%が、訂正ベクトル $[W]$ の計算に用いられる。図4に示す基本復号化の場合、 $[W]$ の計算に用いられるゲート数を10で割る。従って、基本デコーダは、43000ではなく約20500のゲートによって実施される。図5の結果は、直角移相打鍵(QPSK)及び加法性のガウス性ホワイト・ノイズ・チャネルによって変調された場合に得られたものであり、データは4ビットで定量化されている。データ・ベクトル $[R']$ の $p=4$ の再弱成分から構成される $q=16$ のテスト・シーケンスに関して、 $m=4$ の復号化サイクルが適用された。 $2m=8$ の符号ワード探索ステップ中、係数 α_i の連続した値は、0.4、0.4、0.5、0.5、0.6、0.6、0.65、0.65であり、係数 β_i は、一定のままであった。即ち、 $\beta_i=7$ である。曲線 I I には、EP-A-0 694 910 に従って、最適バージョンの復号化プロセスを適用することによって得られる結果が示されている。比較のため、曲線 I には、チャネル符号化がない場合に観測される性能が示されている。基本復号化の単純化によって生じる性能の劣化が、 10^{-5} の BER の場合、0.15 dB 未満に留まるのは明白である。回路の複雑さに関連して、50%の利得と比較考量すると、この劣化はわずかなものである。同じ基本回路を利用して、より多くの回数にわたる反復の実施を企図することが可能であるが、こうした場合、回路全体の複雑さをさらに低減することが必要になる。

【0049】図6には、用いられる基本符号が同じである特定の場合に、上述のようなアルゴリズムに従って積符号の復号化を実施するのに適したBTC回路17のアーキテクチャが示されている。図4による基本復号化ステップは、それぞれ、BTC回路の制御プロセッサ66による制御を受ける、専用の演算及び論理回路65によって実行される(いくつかの回路65を利用して、いくつかの基本復号化を並列に実施することも可能であ

る)。RAMメモリ67は、行列 $\{R\}$ 、 $\{R^{\sim}\}$ 、及び、 $\{D\}$ のサンプルの記憶に用いられる。

【0050】プロセッサ66は、図3の概要フローチャートによる復号化を監視する。信号 $R(t)$ のサンプルを受信すると、プロセッサ66は、入力行列 $\{R\}$ を形成して(ステップ30)、適合するアドレスに記憶し、行列 $\{R^{\sim}\}$ 及び $\{D\}$ を作成して(ステップ31)、適合するアドレスに記憶するため、メモリ67に対する書き込みの順序付けを行う。各基本復号化37または42において、プロセッサは、ベクトル $[R^{\sim}]$ 、 $[D]$ 、及び、 $[R]$ に関する適合するサンプルを基本デコーダ65に供給するため、メモリ67からの読み取り操作の順序付けを行い、次に、これらのベクトル $[R^{\sim}]$ 及び $[D]$ に関する新たな値を記録するため、書き込み操作の順序付けを行う。mサイクルの最後に、プロセッサ66は、ステップ45を実行し、デコーダ17の出力信号 $z(t)$ を送り出すため、メモリ67の適合するアドレス(行列 $\{D\}$)における読み取り操作の順序付けを行う。

【0051】本発明によれば、復号化のために同じBTC回路17を利用しながら、さまざまな積符号パラメータを変更することが可能である。即ち、必要なことは、プロセッサ66のステップ30における入力行列 $\{R\}$ の形成に関連したプログラムの部分に、また、オプションにより、ステップ46における情報ビットの抽出に関連した部分に適合するパラメータが供給されることだけである。

【0052】プログラミングは、図7に用いられる基本符号が同じである特定の場合におけるブロック図が示されているコーダ12に関しても行われる。従来の演算回路68は、伝送すべきビット行列の行及び列の逐次基本符号化に利用される(図2のステップ22及び23)。基本コーダ68は、コーダの制御プロセッサ69によって制御される。RAMメモリ70は、行列 $\{c\}$ のサンプルを記憶するために利用される。

【0053】信号 $X(t)$ の2進サンプルのブロックを受信すると($k=k_1, k_2$ のとき、 $u=1, 2, \dots, k-X$ に関して、ここでは a_u と表示される)、プロセッサ69は、組織符号化のため、行列 $\{c\}$ の部分行列である、 k_1 行及び k_2 列からなる行列 $\{a\}$ を形成するため、メモリ70における書き込み操作の順序付けを行う(ステップ21)。各基本符号化毎に、プロセッサ69は、コーダ68に符号化すべき行または列の適合するサンプルを供給するために、メモリ70における読み取り操作の順序付けを行い、得られた冗長ビットの値を記憶するために、書き込み操作の順序付けを行う。 n_1+n_2 の基本符号化後、最終行列 $\{c\}$ のビットがメモリ70で得られるようになり、プロセッサ69は、変調器に信号 $Y(t)$ を供給するため、このメモリの適合するアドレスにおける読み取り操作の順序付

けを行う。信号 $Y(t)$ の2進サンプルは、 $n=n_1, n_2$ のとき、 $v=1, 2, \dots, n-X-Y$ に関して、ここでは c_v と表示される。

【0054】コーダ及びデコーダのプログラミングによって、積符号に対する短縮技法及び／またはバンクチャリング技法の適用が可能になる。

【0055】短縮の場合、プログラミングは、積符号が適用される行列 $\{a\}$ のビット数 k と符号化すべきブロック当たりのビット a_u の数 $k-X$ との差を表した数 X をコーダ及びデコーダに供給することである。コーダは、この数 X から、各ブロックの処理時には行列 $\{c\}$ の対応する位置に配置され、伝送されるビット c_v からは排除されることになる、決定された値のビット(例えば、0)に関して、行列 $\{a\}$ 内における X の位置を決定する。コーダは、また、各ブロックのビット a_u が、行列 $\{a\}$ の他の位置に記憶されることになる順序も決定する。

【0056】これらの位置を決定するため、コーダ及びデコーダのプロセッサ69、66は、例えば、図8にそのフローチャートが示されている手順のような、既定の手順を適用する。この例の場合、 k_1 行及び k_2 列からなる行列 $\{h\}$ は、 $h_{i,j}$ によって既知ビットの位置 i, j を指定する。当初、指標 i, j 及び n_x である、行列 $\{h\}$ の全ての成分は0である(ステップ80)。指標 n_x は、ステップ81において X と比較され、 $n_x < X$ の場合、指標 i 及び j は、ステップ82において、それぞれ、 k_1 及び k_2 を法として、1ずつインクリメントされる(本書で用いられる表記法の場合、インクリメントが、図8のブロック82に示す公式に従って実施されるように、指標 i 及び j は、それぞれ、1と k_1 の間、及び、1と k_2 の間に含まれる)。ステップ82の後、 $h_{i,j} \neq 0$ の場合(テスト83)、テスト83の反復前に、列指標 j は、ステップ84において k_2 を法として1だけインクリメントされる。テスト83によって、 $h_{i,j} = 0$ であることが分かると、ステップ85において、この成分 $h_{i,j}$ に値1が割り当てられ、比較81ステップに戻る前に、指標 n_x が1だけインクリメントされる。比較ステップ81によって $n_x = X$ であることが分かると、全ての位置が割り当てられる。

【0057】上記手順によって、行列 $\{a\}$ の行及び列における既知ビットの位置の均一な分散が可能になる。 X が k_1 及び k_2 の倍数である場合、均一性は完全である。即ち、全ての行が、全ての列と同じ数の $h_{i,j} = 1$ の位置を備えることになる。そうでない場合、均一性からの偏差が最小限に抑えられる。図9には、 $k_1 = k_2 = 10$ で、 $X = 30$ の(空のボックスは $h_{i,j} = 0$ に対応する)特定の場合における行列 $\{h\}$ の1つの形式が示されている。

【0058】 $h_{i,j} = 1$ の場合における X の位置の決定後、プロセッサ69、66は、それぞれ、符号化すべき

10

20

30

40

50

各ブロックの連続したビット a_u が記憶されることになる、行列 $\{a\}$ の位置に関する行指標及び列指標をそれぞれ与える、2つのテーブル $x(u)$ 、 $y(u)$ ($1 \leq u \leq k-X$)を計算する。これらのテーブルは、ステップ86において、特定の順序で、例えば、1行ずつ ($i = 1, 2, \dots, k_1$)、そして、各行毎に、列指標の昇順に ($j = 1, 2, \dots, k_2$)、 $h_{i,j} \neq 1$ といった行列 $\{a\}$ の位置 i, j を割り当てることによって得られる。

【0059】図8の手順は、コーダ及びデコーダのプログラミング時に1回実施されるので、テーブル $x(u)$ 及び $y(u)$ と行列 $\{h\}$ は、メモリに保持されている。ビット a_u の各ブロック毎に、コーダ12のプロセッサ69は、ステップ21において下記に従って行列 $\{a\}$ を作成する。

$$a_{i,j} = 0 \quad (h_{i,j} = 1 \text{ の場合})$$

$$a_{x(u), y(u)} = a_u \quad (1 \leq u \leq k-X)$$

【0060】その出力信号がブロックに相当する場合、コーダ12には、 $h_{i,j} = 1$ のようなビット $c_{i,j}$ は含まれない(ステップ24)。ステップ30における行列 $\{R\}$ の作成において、デコーダ17のプロセッサ66は、これらの位置に、その符号がビット $a_{i,j}$ の既知の値に相当し(例えば、 $a_{i,j} = 0$ の場合の-1)、その絶対値 M が最高の信頼性(一般に、デコーダの定量化値の最大値)を表すサンプル $R_{i,j}$ を配置する。

【0061】ブロック復号化の最後に(ステップ45)、プロセッサ66は、◎

【数24】

$$\hat{a}_u = D_{x(u), y(u)}$$

に従って、ビット a_u の推定値◎

【数25】

$$\hat{a}_u (= \pm 1)$$

を抽出する。

【0062】バンクチャリングに関して、プログラミングは、積符号(符号の短縮が実施されない場合、 $X=0$)を適用することによって生じる行列 $\{c\}$ の未知のビット数 $n-X$ と各情報ブロック毎にコーダによって送り出されるビット数 $n-X-Y$ の差を表した数 Y をコーダ及びデコーダに供給することである。コーダは、この数 Y 化ら、伝送されるビット c_v から排除されることになるビットに関して、行列 $\{c\}$ における Y の位置を決定する。

【0063】これらの位置を決定するため、コーダ及びデコーダのプロセッサ69、66は、図8における手順と同様の、そのフローチャートが図10に示されている手順のような、既定の手順を適用する。この例の場合、行列 $\{h\}$ の次元は、 n_1 行及び n_2 列に拡張され、バンクチャリングを施されるビットの Y の位置 i, j は、 $h_{i,j} = 2$ によって指定される。当初、 $X \neq 0$ で、指標

i, j 、及び、 n_Y の場合に、図8の手順によって1に設定される成分を除き、行列 $\{h\}$ の全ての成分は0である(ステップ100)。指標 n_Y は、ステップ101において Y と比較され、 $n_Y < Y$ の場合、指標 i 及び j は、ステップ102において、それぞれ、 n_1 及び n_2 を法として1ずつインクリメントされる(本書で用いられる表記法の場合、指標 i 及び j は、それぞれ、インクリメントが図10のブロック102に示す公式に従って実施されるように、1と n_1 の間、及び、1と n_2 の間で構成される)。ステップ102の後、 $h_{i,j} \neq 0$ の場合(テスト103)、テスト103の反復前に、列指標 j は、ステップ104において n_2 を法として1だけインクリメントされる。テスト103によって、 $h_{i,j} = 0$ であることが分かったと、ステップ105において、この成分 $h_{i,j}$ に値2が割り当てられ、比較101ステップに戻る前に、指標 n_X が1だけインクリメントされる。比較ステップ101によって $n_Y = Y$ であることが分かったと、全ての位置が割り当てられる。

【0064】上記手順によって、行列 $\{c\}$ の行及び列におけるバンクチャリングを施されたビットの Y の位置の均一な分散が可能になる。短縮がなければ、 Y が n_1 及び n_2 の倍数である場合、均一性は完全である。そうでないならば、均一性からの偏差が最小限に抑えられる。図11には、短縮化が行われずに、 $n_1 = n_2 = 12$ で、 $Y = 24$ (空のボックスは $h_{i,j} = 0$ に対応する)の場合における行列 $\{h\}$ の1つの形式が示されている。

【0065】短縮がバンクチャリングに関連して用いられると($X \neq 0$ 、 $Y \neq 0$)、行列 $\{c\}$ の行及び列における Y の位置の均一性は、 Y が n_1 の倍数であれば、正方向行列($k_1 = k_2$ 及び $n_1 = n_2$)の場合に完全である。そうでないならば、均一性の偏差は、極めてわずかなままである。図12には、 $k_1 = k_2 = 10$ 、 $n_1 = n_2 = 12$ 、 $X = 30$ 、及び、 $Y = 24$ の場合における行列 $\{h\}$ の1つの形式が示されている。図12の空のボックスに配置される $n-X-Y=90$ のビット $c_{i,j}$ だけが、コーダによって送り出される。

【0066】 $n-X-Y$ のビット c_v の伝送は、例えば、行毎に特定の順序で実施され、従って、ステップ24は、コーダに関して、下記を実施することになる。

$c_v = c_{x'(v), y'(v)}$ $1 \leq v \leq n-X-Y$ の場合
行及び列指標 $x'(v)$ 、 $y'(v)$ は、ステップ106におけるコーダ及びデコーダのプログラミングの際に決定され、記憶される(図10)。相応じて、デコーダ17のプロセッサ66は、ステップ30において、行列 $\{R\}$ の適合する位置に受信信号ブロックの $n-X-Y$ のサンプル R_v を配置する。

$R_{x'(v), y'(v)} = R_v$ $1 \leq v \leq n-X-Y$ の場合
他の位置には、プロセッサ66は、

$R_{i,j} = -M$ (前述のように $h_{i,j} = 1$ であれば)

$R_{i,j} = \pm \varepsilon$ ($h_{i,j} = 2$ であれば)

を配置する。

【0067】数 ε は、Yの推定値における最高の信頼度（一般に、デコーダの定量化値の最低値）を表している。

【0068】反復復号化中、 $h_{i,j} = 1$ のようなビットの推定値 $D_{i,j}$ は、極めて低いままである。バンクチャリングを施されたビット（ $h_{i,j} = 2$ ）に関連した推定値から分かるように、その信頼性は、基本復号化の進行につれて増すことになる。

【0069】2つの同じ基本符号BCH(32, 26, 4)の積、及び、直角移相打鍵(QPSK)による変調の場合に、本発明による積符号に適用される短縮及びバンクチャリング技法の性能が、それぞれ、図13及び14に示されている。積符号のパラメータは、従って、

- ・ $k = 676$
- ・ $n = 1024$
- ・ $d = 16$
- ・ 初期効率 0.660

である。

【0070】図13及び14において、曲線Iは、チャネル符号化が行われない場合の S/N 比Eb/N0の関数としての2進エラー・レート(BER)の進展を示し、積符号に関する曲線IVは、短縮またはバンクチャリング($X=Y=0$)が行われず、反復復号化が $m=4$ サイクルで行われる場合に適用される。

【0071】図13は、 $X=312$ 、 $Y=0$ の場合に相当する、即ち、符号化すべき各ブロックが、 $k-X=364$ 情報ビットから構成され、符号化効率が $(k-X)/(n-X)=0.511$ であり、従って、理論上のシャノン限界は、0.04dBである。曲線V、1~V、4は、それぞれ、 $m=1$ 、 $m=2$ 、 $m=3$ 、及び、 $m=4$ の復号化サイクル後に得られた性能を示している。4サイクル後の曲線の勾配は、ブロック毎に k 情報ビットが伝送される場合（曲線IV）に対してほぼ変化の少ないことが分かる。4サイクル後に $BER=10^{-5}$ を得るのに必要な S/N 比は、2.91dBに等しく、これは、シャノン限界を約2.0dB超える値に相当する。この解決法によって、理論上のシャノン限界に対してコーデックの性能をあまり劣化させることなく、 k_1 以下の任意のサイズのデータ・ブロックを伝送可能なBTC回路が利用可能になる。

【0072】図14は、同じBTC回路が、 $X=0$ 、 $Y=104$ になるようにプログラムされた場合に相当する、即ち、各符号化ブロックが、348ビットではなく、 $n-k-Y$ 冗長ビットから構成され、符号化効率が $k/(n-Y)=0.735$ であり、従って、理論上のシャノン限界は、0.81dBである。曲線V、1~V、4は、それぞれ、バンクチャリングを施されたビットが、組織積符号によって加えられた冗長ビットから選

択される場合の、 $m=1$ 、 $m=2$ 、 $m=3$ 、及び、 $m=4$ の復号化サイクル後に得られた性能を示している。4サイクル後の曲線の勾配は、行列の n ビットが伝送される場合（曲線IV）に対してほぼ変化の少ないことが分かる。4サイクル後に $BER=10^{-5}$ を得るのに必要な S/N 比は、3.71dBに等しく、これは、シャノン限界を約2.9dB超える値に相当する。この解決法によって、理論上のシャノン限界に対してコーデックの性能をあまり劣化させることなく、 $n-k$ 以下の任意の数の冗長ビットを用いることが可能なBTC回路が利用可能になる。

【図面の簡単な説明】

【図1】 本発明によるプロセスの実施に利用可能なデジタル伝送連鎖のブロック図である。

【図2】 積符号の適用を示すフローチャートである。

【図3】 本発明に従って利用可能な反復復号化段階の概要フローチャートである。

【図4】 行または列に関する基本復号化ステップの詳細を示すフローチャートである。

【図5】 図3及び4による反復復号化の性能を示すグラフである。

【図6】 本発明によるチャネル復号化回路及びチャネル符号化回路のそれぞれのブロック図である。

【図7】 本発明によるチャネル復号化回路及びチャネル符号化回路のそれぞれのブロック図である。

【図8】 積符号の短縮に用いることが可能な割り当て手順のフローチャートである。

【図9】 図8の手順の結果を示すダイアグラムである。

【図10】 積符号のバンクチャリングに用いることが可能な割り当て手順のフローチャートである。

【図11】 108の手順の結果を示すダイアグラムである。

【図12】 108の手順の結果を示すダイアグラムである。

【図13】 短縮された積符号及びバンクチャリングを施された積符号に適用される反復復号化の性能を示すグラフである。

【図14】 短縮された積符号及びバンクチャリングを施された積符号に適用される反復復号化の性能を示すグラフである。

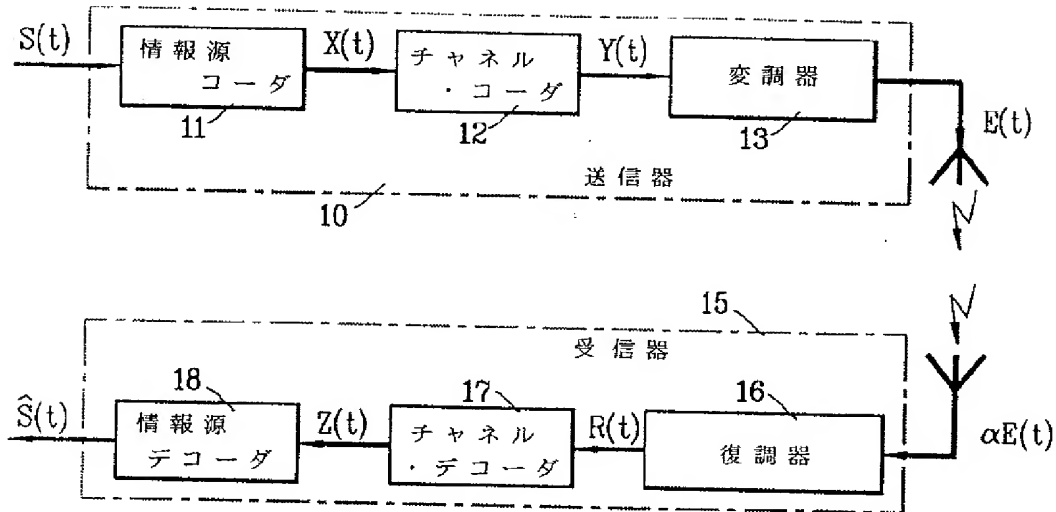
【符号の説明】

- 10……送信器
- 11……情報源コーダ
- 12……チャネル・コーダ
- 13……変調器
- 15……受信器
- 16……復調器
- 17……チャネル・デコーダ
- 18……情報源デコーダ

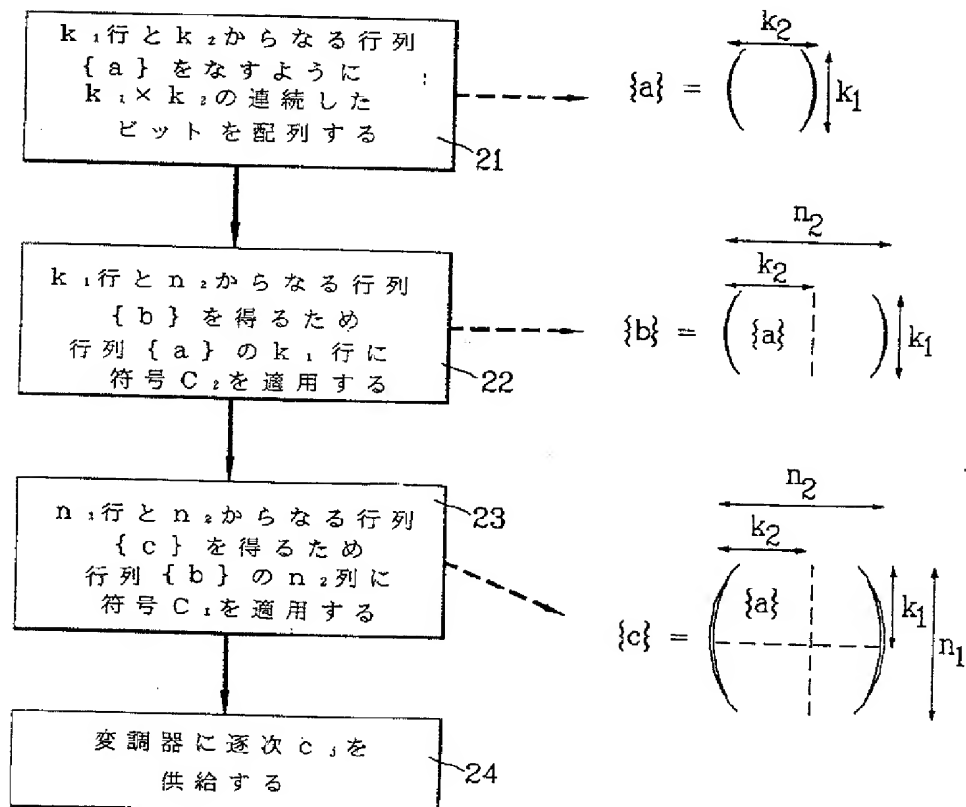
65……基本デコーダ
66, 69……プロセッサ

* 67, 70……メモリ
* 68……基本コーダ

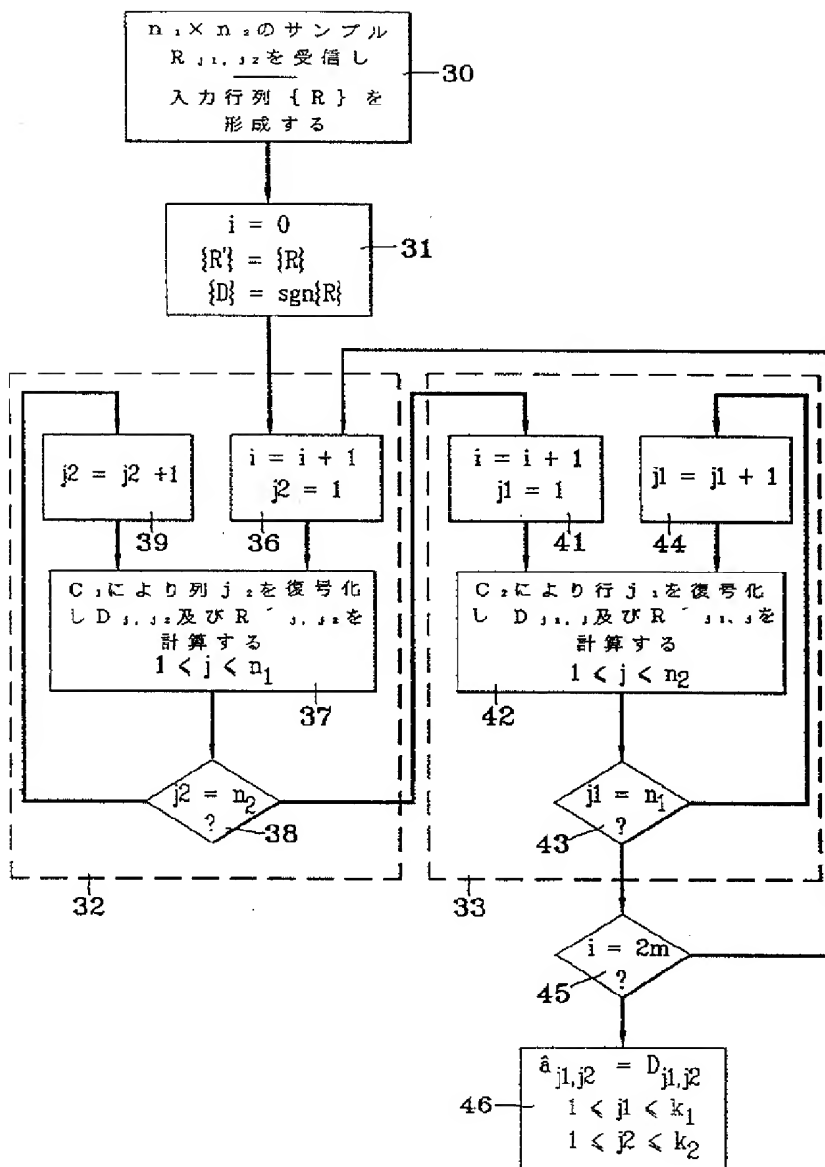
【図1】



【図2】



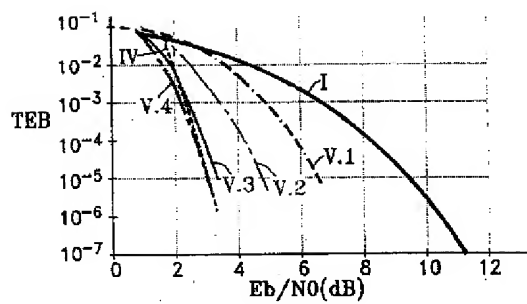
【図3】



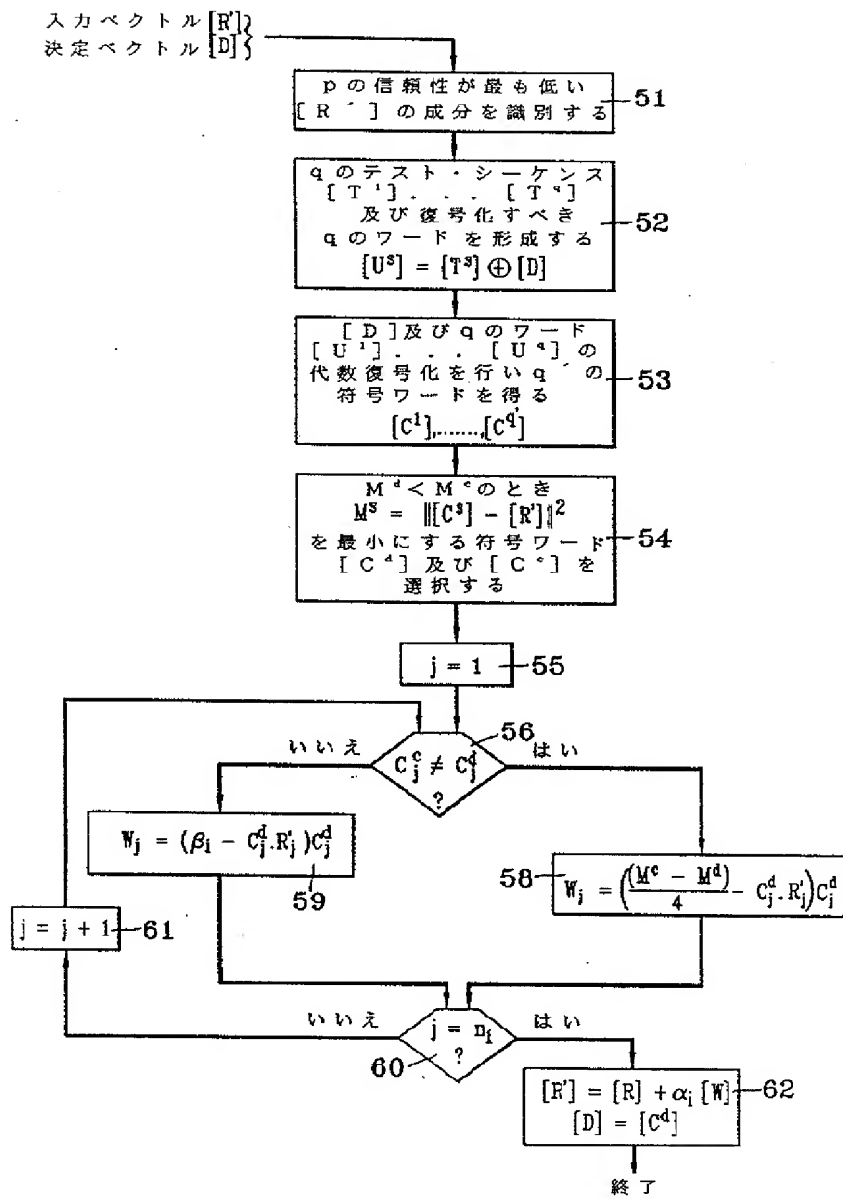
【図9】

		j									
		1	2	---	---	---	---	---	---	k ₂	
i	1	1	1	1							
	2		1	1	1						
				1	1	1					
					1	1	1				
						1	1	1			
							1	1	1		
								1	1	1	
									1	1	1
										1	1
	k ₁	1	1								1

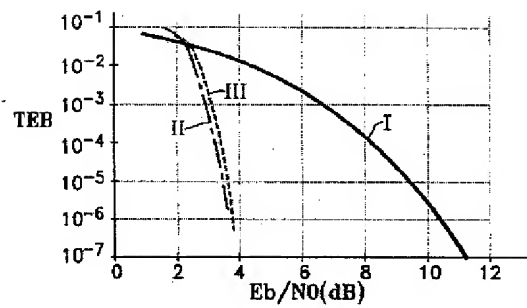
【図13】



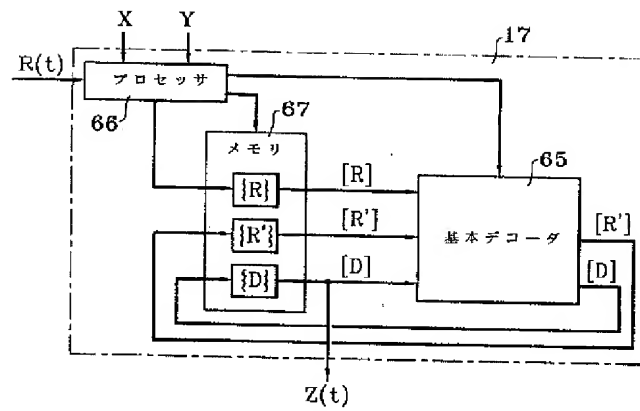
【図4】



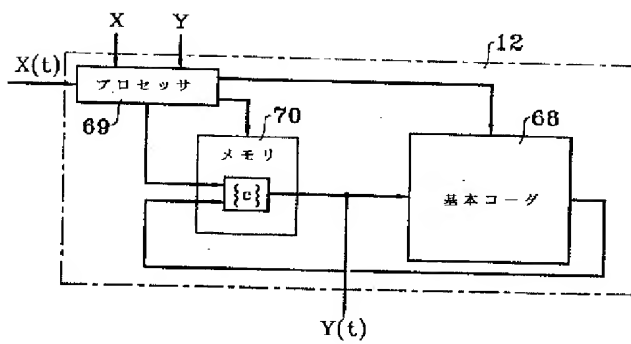
【図5】



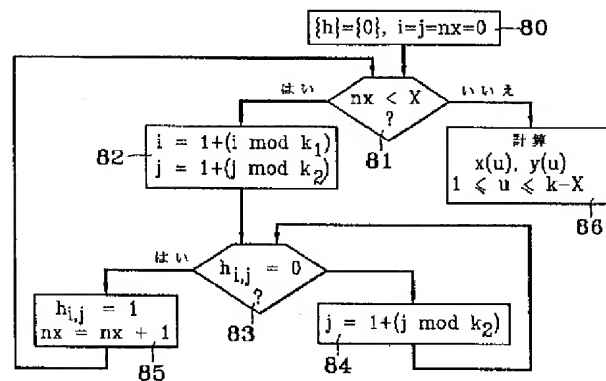
【図6】



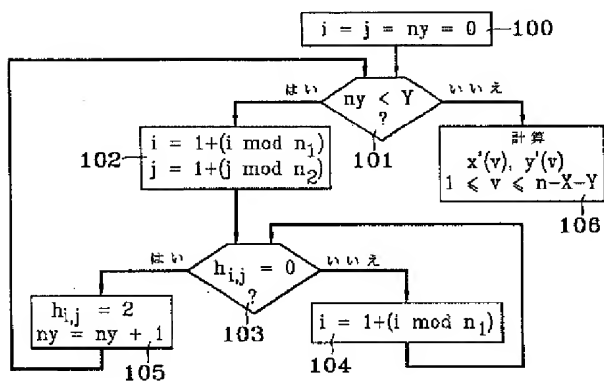
【図7】



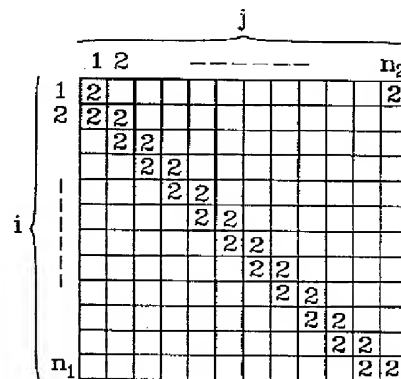
【図8】



【図10】



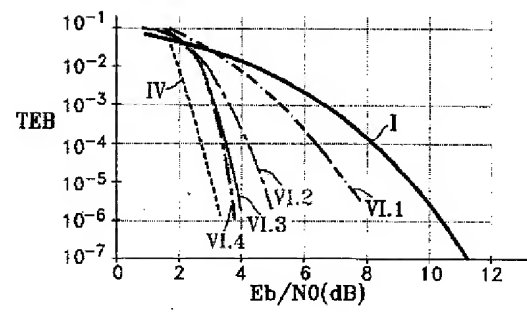
【図11】



【図12】

		j														
		1	2	-----								k_2	n_2			
i {	1	1	1	1										2	2	
	2	2	1	1	1									2	2	
		2	2	1	1	1										
			2	2	1	1	1									
				2	2	1	1	1								
					2	2	1	1	1							
						2	2	1	1	1						
							2	2	1	1	1					
								2	2	1	1	1				
									2	2	1	1	1			
										2	2	1	1	1		
	k_1	1	1						2	2	1	1				
	n_1										2	2		2	2	

【図14】



【外国語明細書】

1 Title of Invention

PROCESS FOR TRANSMITTING INFORMATION BITS WITH ERROR
CORRECTION CODING, CODER AND DECODER FOR THE
IMPLEMENTATION OF THIS PROCESS

2 Claims

1. Process for transmitting information bits, in which a transmitter (10) executes the following steps:

- formation of a first binary matrix ($\{a\}$) from the information bits to be transmitted (a_0);

- conversion of the first binary matrix into a second binary matrix ($\{c\}$) by application of a systematic block code corresponding to the product of elementary systematic block codes (c_1, c_2); and

- transmission of the bits (c_v) extracted from the second binary matrix towards a channel,

and a receiver (15) executes the following steps:

- formation of an input matrix ($\{R\}$) from a signal ($R(t)$) received according to the said channel, the input matrix being of the same size as the second binary matrix, and being comprised of digital samples the signs of which represent respective initial estimations of the bits in the second binary matrix and the absolute values of which respectively measure the confidences associated with the said initial estimations; and

「以下 余白」

- iterative decoding of the input matrix including a number m of decoding cycles, each decoding cycle successively comprising code word search steps (32,33) for each elementary block code used in the product code,

in which, at each code word search step (32,33), a data matrix $\{R'\}$ and a decision matrix $\{D\}$ with binary components are received which, before the first search step for the iterative decoding, are respectively constituted by the input matrix $\{R\}$ and by the matrix the binary components of which correspond to the signs of the input matrix samples, and for the following search step a new decision matrix $\{D\}$ is produced, the binary components of which represent new respective estimations of the bits of the second binary matrix and a new data matrix $\{R'\}$ the samples of which have absolute values

「以下 余白」

which respectively measure confidences associated with the said new estimations,

in which decoded information bits (\hat{a}_u) are extracted from the decision matrix produced during the last code word search step,

and in which each code word search step (32,33) includes a division of the data matrix received into data vectors ($[R']$) each corresponding to a code word of the elementary code and a corresponding division of the decision matrix received into decision vectors ($[D]$) and elementary decodings with soft decisions (37,42) to respectively process at least certain data vector/decision vector pairs.

characterised in that the first binary matrix ($\{a\}$) comprises, in addition to the information bits (a_u), a set of bits with values a priori known to the receiver, which are distributed in an approximately uniform manner according to each dimension of the first binary matrix, which, after systematic coding, are located in determined positions of the said second binary matrix ($\{c\}$), and which are not transmitted towards the channel, and that the receiver (15) places in the input matrix ($\{R\}$), in positions corresponding to the said determined positions of the second binary matrix, samples the signs of which correspond respectively to the a priori known values of the bits of the said set and the absolute values of which are representative of a maximum confidence.

2. Process according to claim 1, characterised in that another set of bits having other determined positions in the second binary matrix ($\{c\}$) are not transmitted towards the channel, and in that the receiver (15) places in the input matrix ($\{R\}$), in positions corresponding to the said other determined positions of the second binary matrix, samples the absolute values of which are representative of minimum confidence.

3. Process according to claim 2, characterised in

- 4 -

that the said other determined positions are distributed approximately uniformly according to each dimension of the second binary matrix $\{C\}$.

4. Process according to claim 1, 2 or 3, characterised in that the soft decision elementary decoding (37,42) for the processing of a data vector/decision vector pair comprises the following steps:

- determination of a number p of indexes (r_1, \dots, r_p) for which the data vector components $\{R'\}$ are the least reliable;

- construction of a number q of binary words to be decoded $\{U^p\}$ from the said p indexes and from the decision vector $\{D\}$;

- obtaining q' code words $\{C^s\}$ on the basis of algebraic decodings of the decision vector and the q binary words to be decoded;

- selection, from the q' code words obtained, of that $\{C^d\}$ having the shortest euclidean distance with the data vector $\{R'\}$;

- calculation of a correction vector $\{W\}$, each component W_j of the correction vector being respectively calculated by determining an optional concurrent word $\{C^c\}$ having its j -th component different from that of the selected code word $\{C^d\}$, by applying the formula:

$$W_j = \left(\frac{|M^c - M^d|}{4} - C_j^d \cdot R_j' \right) C_j^d$$

when a concurrent word has been determined, M^d and M^c respectively designating the euclidean distances, with respect to the data vector $\{R'\}$, of the selected code word $\{C^d\}$ and of the concurrent word $\{C^c\}$, and C_j^d and R_j' respectively designate the j -th components of the selected code word and of the data vector;

- obtaining the new decision vector ([D]) taken as equal to the said selected code word ([C^d]); and

- calculation of the new data vector ([R']) by adding the correction vector ([W]) multiplied by a first confidence coefficient (α_i) to the corresponding input vector ([R]) extracted from the input matrix ([R]).

5. Process according to claim 4 characterised in that, in the calculation step for a correction vector ([W]), the determination of an optional concurrent word ([C^c]) with respect to the j-th component of the selected code word ([C^d]) includes a comparison between the j-th component of the selected code word ([C^d]) and that of a candidate code word which, among the q' code words obtained except for the selected code word, has the shortest euclidean distance with the data vector ([R']), the said candidate code word being taken as a concurrent word when its j-th component is different from that of the selected code word, and no concurrent word being determined if not.

20 6. Process according to claim 4 or 5, characterised in that the calculation of each component W_j of the correction vector ([W]) in the absence of determination of a concurrent word with respect to that component is carried out according to the formula:

$$W_j = \beta_i C_j^d$$

25 where β_i designates a second confidence coefficient.

7. Process according to claim 4 or 5, characterised in that the calculation of each component W_j of the correction vector ([W]) in the absence of determination of a concurrent word with respect to that component is carried out according to the formula:

30

$$W_i = (\beta_i - C_i^a R_j^a) C_j^a$$

where β_i designates a second confidence coefficient.

8. Redundancy coder (12), comprising:

- means (69) to form a first binary matrix $\{a\}$ from information bits to be transmitted (a_i) ;
- elementary coding means (68) instructed to convert the first binary matrix into a second binary matrix $\{c\}$ by application of a systematic block code corresponding to the product of elementary systematic block codes (C_1, C_2) ; and

- means to transmit the bits (c_i) extracted from the second binary matrix towards a channel,

characterised in that it also comprises programming means (69) in order to determine, from a number X supplied to program the number of information bits to be included in each first matrix, X positions uniformly distributed according to each dimension of the first matrix, and in that the means for creating the first matrix are designed to place bits of known values in the said X positions, the said bits of known values being located after systematic coding in determined positions of the second binary matrix and not being transmitted towards the channel.

9. Coder according to claim 8, characterised in that the programming means (69) are designed to determine, from a number Y supplied in order to program the number of bits of the second matrix transmitted towards the channel, Y positions in the second matrix $\{c\}$, and in that the bits of the second matrix located at the said Y positions are not transmitted towards the channel.

10. Coder according to claim 9, characterised in that the programming means (69) distribute the said Y

positions in an approximately uniform manner according to each dimension of the second binary matrix ($\{c\}$).

11. Error correction decoder (17), comprising:

- means (66) for forming an input matrix ($\{R\}$)
5 from a signal $R(t)$ received according to a transmission channel, the input matrix being composed of digital samples the signs of which represent respective initial estimations of bits of a binary matrix formed by a redundancy coder (12) applying a systematic block code
10 corresponding to the product of elementary systematic block codes (c_1, c_2) and the absolute values of which respectively measure the confidences associated with the said initial estimations; and

- iterative decoding means (65) instructed to
15 decode the input matrix according to successive decoding cycles, each decoding cycle successively comprising steps (32,33) for searching code words for each elementary block code used in the product code,

in which, at each code word search step (32,33),
20 the decoding means (65) receive a data matrix ($\{R'\}$) and a decision matrix ($\{D\}$) with binary components which, before the first search step for the iterative decoding, are respectively constituted by the input matrix ($\{R\}$) and a matrix the binary components of which correspond to the
25 signs of the input matrix ($\{R\}$) samples, and for the following search step a new decision matrix ($\{D\}$) is produced, the binary components of which represent new respective estimations of the bits of the second binary matrix and a new data matrix ($\{R'\}$) the samples of which
30 have absolute values which respectively measure confidences associated with the said new estimations, decoded information bits (\hat{a}_u) being extracted from the decision matrix produced during the last step of the code word search,

35 and in which each code word search step (32,33) includes a division of the data matrix received into data

vectors ($[R']$) each corresponding to a code word of the elementary code and a corresponding division of the decision matrix received into decision vectors ($[D]$) and elementary decodings with soft decisions (37,42) to
5 respectively process at least certain data vector/decision vector pairs,

characterised in that it further comprises programming means (66) to determine, from a number X supplied to program the number of decoded information bits
10 to be extracted from the decision matrix, X positions uniformly distributed according to each dimension of a non-redundant sub-matrix ($\{a\}$) of the said binary matrix, and in that the means for forming the input matrix ($\{R\}$) are designed to place, in positions corresponding to the
15 said X positions, digital samples of determined signs and the absolute values of which are representative of maximum confidence.

12. Decoder according to claim 11, characterised in that the programming means (66) are designed to
20 determine, from a number Y supplied in order to program the number of samples of the input matrix obtained from the signal received, Y positions in the input matrix ($\{R\}$), and in that the means of forming the input matrix are designed to place in the said Y positions digital
25 samples the absolute values of which are representative of maximum confidence.

13. Decoder according to claim 12, characterised in that the programming means (66) distribute the said Y positions in an approximately uniform manner according to
30 each dimension of the input matrix ($\{R\}$).

「以下 余白」

3 Detailed Description of Invention

The present invention relates to a process for transmitting information bits, in which a transmitter executes the following steps:

- formation of a first binary matrix from the information bits to be transmitted;

- conversion of the first binary matrix into a second binary matrix by application of a systematic block code corresponding to the product of elementary systematic block codes; and

- transmission of the bits extracted from the second binary matrix towards a channel,

and a receiver executes the following steps:

- formation of an input matrix from a signal received according to the said channel, the input matrix being of the same size as the second binary matrix, and being comprised of digital samples the signs of which represent respective initial estimations of the bits in the second binary matrix and the absolute values of which respectively measure the confidences associated with the said initial estimations; and

- iterative decoding of the input matrix including a number m of decoding cycles, each decoding cycle successively comprising code word search steps for each elementary block code used in the product code,

in which, at each code word search step, a data matrix and a decision matrix with binary components are received which, before the first search step for the iterative decoding, are respectively constituted by the input matrix and by the matrix the binary components of which correspond to the signs of the input matrix samples, and for the following search step a new decision matrix is

「以下 余白」

produced, the binary components of which represent new
respective estimations of the bits of the second binary
matrix and a new data matrix the samples of which have
absolute values which respectively measure confidences
5 associated with the said new estimations,

in which decoded information bits are extracted
from the decision matrix produced during the last code
word search step,

and in which each code word search step includes a
10 division of the data matrix received into data vectors
each corresponding to a code word of the elementary code
and a corresponding division of the decision matrix
received into decision vectors and elementary decodings
with soft decisions to respectively process at least
15 certain data vector/decision vector pairs.

On the subject of digital transmission of
information (speech, image, data, etc.), a distinction is
usually made between source coding and channel coding.
Source coding forms the binary representation of the
20 signal to be transmitted. It is normally designed as a
function of the nature of the signal to be transmitted.
Much effort has been expended in recent years on the
subject of source coding in order to reduce the digital
rate, while preserving good transmission quality.
25 However, these new source coding techniques require better
protection of the bits against perturbations during the
transmission. Moreover, the physical and economic
limitations of high-frequency components (noise factor,
power saturation), as well as regulations on the level of
30 power allowed for transmission limit the range of digital
transmission systems.

For this reason much work has been carried out on
the subject of channel coding, in particular on the
subject of block coding. This type of error-correcting
35 coding consists in adding $n-k$ redundancy bits to k
information bits originating from the source coding, and

in using these redundancy bits on reception in order to correct certain transmission errors. The ratio $R=k/n$ is called the efficiency of the code, and the coding gain G is defined as the ratio, expressed in decibels, between the energies per information bit E_b which are necessary at the input of the receiver without coding and with coding in order to reach a given binary error rate (BER). A typical objective is to create coders and especially their associated decoders such that: (i) the coding gain G is as high as possible ($G > 5$ dB for $BER = 10^{-5}$), (ii) the code efficiency R is as high as possible ($R > 0.6$), and (iii) the complexity of the decoding is as low as possible.

The case of the storage of the digital information may be seen as a particular case of transmission, in which the propagation channel includes a memory where the information remains in more or less long-term storage, the transmitter and the receiver may be the same or not. It will thus be understood that, in general, the notions of channel coding and of associated decoding are applicable to the field of the storage of information in the same way as to transmission, the errors to be corrected then being those due to the reading or to the writing in the memory, to the alteration in the content of the memory or also to communications (remote or not) with the devices for reading and writing in the memory.

It is known to enhance the performances of error-correcting codes by using concatenation techniques. In particular, the technique of product codes, which is more particularly involved with the present invention, makes it possible, from two simple block codes (that is to say having a small minimum Hamming distance d) to obtain a code whose minimum Hamming distance is equal to the product of the Hamming distances of the elementary codes used (see S.M. Reddy: "On decoding iterated codes", IEEE Trans. on Information theory, Vol. IT-16, No. 5, September 1970, pages 624-627).

If a block code with parameters (n_1, k_1, d_1) is designated by C_1 and a block code with parameters (n_2, k_2, d_2) is designated by C_2 , the application of the code which is the product of C_1 with C_2 consists in ordering the
5 $k_1 \times k_2$ successive information bits in a matrix, and in coding the k_1 rows of the matrix by the code C_2 , then the n_2 columns of the resultant matrix by the code C_1 . The parameters of the product code P are then given by $(n = n_1 \times n_2; k = k_1 \times k_2; d = d_1 \times d_2)$. The efficiency R of the
10 code P is equal to $R_1 \times R_2$. Decoding of the code P according to the maximum likelihood a posteriori (MLP) makes it possible to reach optimal performance. The maximum asymptotic coding gain can then be approximated by the relation $G < 10 \log_{10} (R \cdot d)$.

15 The product code is thus very beneficial, but decoding according to the MLP is generally too complex, except in the case of short block codes.

In their article "Separable MAP filters for the decoding of product and concatenated codes", Proc. ICC'93,
20 Geneva, Pages 1740-1745, May 1993, J. Lodge et al. proposed an iterative decoding algorithm of the type indicated in the introduction, in which the row vectors and the column vectors extracted from the data matrix are decoded using Bahl's algorithm (see L.R. Bahl et al,
25 "Optimal decoding of linear codes for minimizing symbol error rate", IEEE Trans. on Information Theory, Vol. IT-20, pages 248-287, March 1974) which estimates Log Likelihood Ratios (LLRs) for bits. Bahl's algorithm provides soft decisions expressed by the LLRs, allowing
30 performances close to those of the MLP to be achieved. However, it makes use of a decoding trellis having a number of statuses which grows exponentially as a function of $n-k$. As a result, while the algorithm of Lodge et al. is suitable for short codes such as, for example,
35 Hamming's code $(16, 11, 3)$, it proves unusable in practice for codes with a high efficiency such as, for example, the

BCH code (63,51,5).

Another process of the type indicated at the beginning was presented in European Patent Application No. 0 654 910, the contents of which are incorporated in the present description.

The latter process can be used to decode all the product codes which are built from codes in linear blocks and for which an algebraic decoder is available. The performances obtained with this process are near optimum (see R. Pyndiah et al.: "Near optimum decoding of product codes", Proc. IEEE GLOBECOM'94 Conference, Vol. 1/3, Nov.-Dec. 1994, San Francisco, pages 339-343). In fact, it can be used to obtain, for a given product code and in four iterations, a BER equal to 10^{-5} for a signal-to-noise ratio in the region of 2.5 dB above the theoretical Shannon limit for the product code in question. Moreover, this process is far less complex and much more reliable than the solution proposed by Lodge et al.. It is thus possible to decode very large product codes, with elementary code lengths n_1, n_2 of up to 256.

In considering the creation of a circuit suitable for carrying out iterative decoding of the type mentioned in the introduction, hereafter called block turbo-code circuit or BTC circuit (see O. Raoul et al., "Architecture and design of a turbodecoder circuit for product codes", Proc. GRETSI'95 Conference, Vol. 2, September 1995, pages 981-984), it is apparent that the surface area of the circuit can be considerably reduced by using the same elementary decoder to carry out several iterations rather than cascading several elementary decoders. Depending on the envisaged application, the number of iterations will be a function of the elementary decoder's complexity. The less the elementary decoder is complex, the greater the number of iterations, hence the advantage of reducing the complexity of the elementary decoder.

For reasons of economy, it is desirable to have a

programmable BTC circuit allowing processing of data blocks of differing sizes with a varying number of redundancy bits. It can thus be envisaged to use the same BTC circuit for different applications, which allows an appreciable saving in terms of development costs.

A principal aim of the present invention is to respond to this requirement by proposing a transmission process for information bits allowing the use of a programmable BTC circuit.

The invention therefore proposes in a process of the type indicated in the introduction, that the first binary matrix comprises, in addition to the information bits, a set of bits with values a priori known to the receiver, which are distributed in an approximately uniform manner according to each dimension of the first binary matrix, which, after systematic coding, are located in determined positions of the said second binary matrix, and which are not transmitted towards the channel, and that the receiver places in the input matrix, in positions corresponding to the said determined positions of the second binary matrix, samples the signs of which correspond respectively to the a priori known values of the bits of the said set and the absolute values of which are representative of a maximum confidence.

The invention makes use of a technique similar to shortening techniques which are well known in the field of simple block codes. Let n , k and d be the product code parameters, in the form:

$$n = \prod_{i=1}^L n_i, \quad k = \prod_{i=1}^L k_i, \quad \text{and} \quad d = \prod_{i=1}^L d_i$$

where L is the number of elementary codes the respective parameters of which are (n_i, k_i, d_i) (the case where $L=2$ is considered below, without limiting generality). k and n

are the respective numbers of bits in the "first" and "second" binary matrices.

The invention allows the number of independent information bits contained in the matrix to be adapted to any number $k-X$ less than or equal to k , the receiver decoding circuit being the same whatever the number X of a priori known bits. The positions of these X bits are uniformly distributed in the first matrix, which allows optimum usage of the performance of the iterative decoding process. On this point, it should be noted that a shortening of one or more codes into elementary blocks would be less advantageous as it would allow less choice in the value of X , and especially as it would lead to certain elementary decodings resulting in no BER gain.

The parameters (n', k', d') of the shortened product code are finally $n'=n-X$, $k'=k-X$ and $d'=d$. Its efficiency R' is $R'=(k-X)/(n-X) \leq k/n$.

In an advantageous implementation, another set of Y bits of determined positions in the second binary matrix are not transmitted towards the channel, and the receiver places in the input matrix, in positions corresponding to these Y determined positions of the second binary matrix, samples the absolute values of which are representative of minimum confidence.

A puncturing technique is used here which is similar to those frequently used in the field of convolutional codes. The puncturing increases the efficiency of the code. In the case of convolutional codes, its aim is generally to achieve coding efficiencies in excess of $1/2$ while using binary codes, i.e. those which have the least complex decoding trellis. In general, a punctured convolutional code has distance properties similar to those of a non-punctured code of identical efficiency.

Puncturing, however, is not normally applied to block codes. In fact, there are numerous block codes with

high efficiencies having optimum distance properties. It would therefore be expected that puncturing would degrade distance properties without achieving a gain in complexity which would be as appreciable as in the case of
 5 convolutional codes. The inventors were surprised to observe that, in the case of a product code, puncturing applied as indicated above, combined with the soft decision iterative decoding process, does not significantly degrade the performance of the codec.

10 The Y parameter can then be used to program the number of bits per coder output block, and the global coding efficiency. For example, if the programming of the number of non-redundant information bits per block (via the number X) leads to an excessive drop in coding
 15 efficiency, a higher efficiency can be achieved if $Y > 0$.

The parameters (n'', k'', d'') of the shortened and punctured product code are finally $n'' = n - X - Y$, $k'' = k - X$ and $d'' = d$. Its efficiency R'' is $R'' = (k - X) / (n - X - Y)$.

The soft decision elementary decoding for
 20 processing of a data vector/decision vector pair can in particular be of the type described in EP-A-0 654 910. It then comprises the following steps:

- determination of a number p of indexes for which the data vector components are the least reliable;
- 25 - construction of a number q of binary words to be decoded from the said p indexes and from the decision vector;
- obtaining q' code words on the basis of algebraic decodings of the decision vector and the q
 30 binary words to be decoded;
- selection, from the q' code words obtained, of that having the shortest euclidean distance with the data vector;
- calculation of a correction vector, each
 35 component W_i of the correction vector being respectively calculated by determining an optional concurrent word

having its j-th component different from that of the selected code word, applying the formula:

$$W_j = \left(\frac{|M^c M^d|}{4} C_j^d \cdot R_j' \right) C_j^d$$

when a concurrent word has been determined, M^d and M^c respectively designating the euclidean distances, with respect to the data vector, of the selected code word and of the concurrent word, and C_j^d and R_j' respectively designate the j-th components of the selected code word and of the data vector;

- obtaining the new decision vector taken as equal to the said selected code word; and

- calculation of the new data vector by adding the correction vector multiplied by a first confidence coefficient to the corresponding input vector extracted from the input matrix.

In a preferred implementation, in the calculation step for a correction vector, the determination of an optional concurrent word with respect to the j-th component of the selected code word includes a comparison between the j-th component of the selected code word and that of a candidate code word which, among the q' code words obtained except for the selected code word, has the shortest euclidean distance with the data vector, the said candidate code word being taken as a concurrent word when its j-th component is different from that of the selected code word, and no concurrent word being determined if not.

A second aspect of the invention relates to a programmable redundancy coder, comprising:

- means to form a first binary matrix from information bits to be transmitted;

- elementary coding means instructed to convert the first binary matrix into a second binary matrix by application of a systematic block code corresponding to

the product of elementary systematic block codes; and

- means to transmit the bits extracted from the second binary matrix towards a channel,

the coder also comprising programming means in order to determine, from a number X supplied to program the number of information bits to be included in each first matrix, X positions uniformly distributed according to each dimension of the first matrix, the means for creating the first matrix being designed to place bits of known values in the said X positions, the said bits of known values being located after systematic coding in determined positions of the second binary matrix and not being transmitted towards the channel.

A third aspect of the invention relates to a programmable error correction decoder, comprising:

- means for forming an input matrix from a signal received according to a transmission channel, the input matrix being composed by digital samples the signs of which represent respective initial estimations of bits of a binary matrix formed by a redundancy coder applying a systematic block code corresponding to the product of elementary systematic blocks and the absolute values of which respectively measure the confidences associated with the said initial estimations; and

- iterative decoding means instructed to decode the input matrix according to successive decoding cycles, each decoding cycle successively comprising steps for searching code words for each elementary block code used in the product code,

in which, at each code word search step, the decoding means receive a data matrix and a decision matrix with binary components which, before the first search step, are respectively constituted by the input matrix and by a matrix the binary components of which correspond to the signs of the samples of the input matrix, and produce for the following search step a new decision matrix the

binary components of which represent respective new estimations of the bits of the second binary matrix and a new data matrix the samples of which have absolute values which respectively measure the confidences associated with
5 the said new estimations, the decoded information bits being extracted from the decision matrix produced during the last code word search step.

and in which each code word search step includes a division of the data matrix received into data vectors
10 each corresponding to a code word of the elementary code and a corresponding division of the decision matrix received into decision vectors and elementary decodings with soft decisions to respectively process at least certain data vector/decision vector pairs,

15 the decoder further comprising programming means to determine, from a number X supplied to program the number of decoded information bits to be extracted from the decision matrix, X positions uniformly distributed according to each dimension of a non-redundant sub-matrix
20 of the said binary matrix, the means for forming the input matrix being designed to place, in positions corresponding to the said X positions, digital samples of determined signs and the absolute values of which are representative of maximum confidence.

25 Other features and advantages of the present invention will be apparent from the description below of non-limitative embodiment examples, to be read in conjunction with the annexed drawings, in which:

- Figure 1 is a block diagram of a digital
30 transmission chain which can be used to implement the process according to the invention;

- Figure 2 is a flowchart illustrating the application of a product code;

- Figure 3 is a general flowchart of an iterative
35 decoding phase which can be used according to the invention;

- Figure 4 is a flowchart detailing an elementary decoding step for a row or a column;

- Figure 5 is a graph illustrating the performance of the iterative decoding according to Figures 3 and 4;

5 - Figures 6 and 7 are respective block diagrams of a channel decoding circuit and a channel coding circuit according to the invention;

10 - Figure 8 is a flowchart of an allocation procedure which can be used for the shortening of a product code;

- Figure 9 is a diagram illustrating the results of the procedure in Figure 8;

15 - Figure 10 is a flowchart of an allocation procedure which can be used for the puncturing of a product code;

- Figures 11 and 12 are diagrams illustrating the results of the procedure in Figure 10; and

20 - Figures 13 and 14 are graphs illustrating the performances of the iterative decoding applied to a shortened product code and to a punctured product code.

The inventors have developed an advantageous variant, providing a good performance/complexity compromise, of the iterative decoding processes for product codes described in EP-A-0 654 910. This variant
25 is described below with reference to Figures 1 to 5, before a more specific description of the structure of a product code according to the invention is given. It can be seen on the one hand that the said variant can be applied to the iterative decoding of any type of product
30 code and on the other hand that the transmission process according to the invention is compatible with other iterative decoding methods, such as for example those described in EP-A-0 654 910 or also in the aforementioned article by J. Lodge et al.

In the transmission chain illustrated in Figure 1, the information bits to be transmitted a_j are contained in a signal

$$X(t) = \sum_j a_j h(t - jT)$$

addressed in input to the channel coder 12 of the transmitter 10. This signal $X(t)$ is formed by the source coder 11 from an analog signal $S(t)$. The source coder 11 is conventionally such that the a_j s are independent and take the value 0 or 1 with equal probability. $h(t)$ designates a time gate of duration T which is the time interval separating two successive bits. The channel coder 12 applies block coding to produce a signal

$$Y(t) = \sum_j c_j h(t - jT')$$

where the c_j are the coded bits and T' is the time interval separating two coded bits ($T' < T$). The modulator 13 converts the sequence $Y(t)$ into signal sequences which are compatible with the propagation channel. In the case of two-state phase shift keying associated with a radio channel, an example of the signal sent is given by:

$$E(t) = \sum_j e_j h(t - jT_s) \sin(2\pi f_0 t)$$

where f_0 is the frequency of the carrier wave and $e_j = 2.c_j - 1$. The signal received at the antenna of the

receiver 15 is attenuated by a coefficient α . The demodulator 16 derives the probability ratio for each bit, which may be expressed:

$$5 \quad R_j = c_j + B_j$$

where the samples B_j are the noise samples induced by the propagation channel, which are independent of the bits c_j , and not correlated with each other, with an average of 0
10 and with standard deviation σ , depending on the signal-to-noise ratio. The signal at the output of the demodulator 16 is then equal to:

$$R(t) = \sum_j R_j h(t - jT)$$

The channel decoder 17 next takes the decision
15 relating to the bits sent by taking advantage of the channel coding used on sending in order to minimize the errors. Its output signal is given by:

$$Z(t) = \sum_j \hat{a}_j h(t - jT)$$

where the bits \hat{a}_j are the decisions taken by the channel decoder. The source decoder 18 next reconstructs the
20 analog signal $S(t)$ from the bits supplied by the channel decoder 17.

The invention resides principally in the channel coder 12 and the channel decoder 17. Thus it will be understood that it is compatible with various types of
25 source coding/decoding, of modulation/demodulation and of propagation channels. In particular, the invention can be applied in the context of digital television. The coder 11 and the decoder 18 can then be produced according to a

MPEG (moving picture expert group) standard, for example, and the modulator 13 and the demodulator 16 are matched to the propagation channel used (radio, wire, etc.). Another application example is facsimile transmission.

5 The block code applied by the channel coder 12 is a product code obtained from systematic elementary codes. In the embodiment described below, it is the product of two linear block codes C_1 , C_2 with respective parameters (n_1, k_1, d_1) and (n_2, k_2, d_2) .

10 The coding procedure, carried out with conventional coding circuits, is illustrated in Figure 2.

The bits a_i received successively from the source coder 11 are first of all held, by groups of $k_1 \times k_2$ bits, according to a matrix $\{a\}$ with k_1 rows and k_2 columns (step 21). The
15 block code C_2 is then applied to each of the k_1 rows of the matrix $\{a\}$, which supplies a matrix $\{b\}$ with k_1 rows and n_2 columns (step 22). As the code C_2 is systematic, k_2 of the n_2 columns of the matrix $\{b\}$ are identical to the matrix $\{a\}$, for example the first k_2 columns. Next (step 23),
20 the block code C_1 is applied to each of the n_2 columns of the matrix $\{b\}$, which supplies a matrix $\{c\}$ with n_1 rows and n_2 columns, the c_j components of which are the bits transmitted successively to the modulator 13 in the form of the signal $Y(t)$ (step 24). As the code C_1 is
25 systematic, k_1 of the n_1 rows of the matrix $\{c\}$ are identical to the matrix $\{b\}$, for example the first k_1 rows. Thus, the upper left part, of k_1 rows and k_2 columns, of the matrix $\{c\}$ is identical to the matrix $\{a\}$, the other components of the matrix $\{c\}$ being redundancy
30 bits. All the columns of the matrix $\{c\}$ are code words of the code C_1 . Likewise, all the rows of the matrix $\{c\}$ are code words of the code C_2 , given that the elementary codes are linear.

35 The channel decoder 17 applies an iterative decoding procedure, the general flow chart for which is presented in Figure 3. After receiving a block of $n_1 \times n_2$

samples R_{j_1, j_2} ($1 \leq j_1 \leq n_1$, $1 \leq j_2 \leq n_2$) of the signal $R(t)$ received from the demodulator 16, which corresponds to the sending of a coded block formed by the channel coder 12, these samples are held in an input matrix $\{R\}$ with n_1 rows and n_2 columns (step 30).

The decoding of this block of $n_1 \times n_2$ samples is initialized (step 31) by initializing the counting variable i to 0, by forming a data matrix $\{R'\}$ with n_1 rows and n_2 columns, the components of which are initially the same as those of the input matrix $\{R\}$, and by forming a decision matrix $\{D\}$ with n_1 rows and n_2 columns, the components of which are binary (-1 or +1) and, initially, each represent the sign of the corresponding component of the input matrix $\{R\}$: $D_{j_1, j_2} = \text{sgn}(R_{j_1, j_2}) = \pm 1$.

After this initialization, the iterative decoding includes a number m of decoding cycles. Each decoding cycle successively includes a step 32 of search for words of the code C_1 in the columns of the data matrix, and a step 33 of search for words of the code C_2 in the rows of the data matrix.

At each search step 32 or 33, new values of the components of the decision matrix $\{D\}$ and of the data matrix $\{R'\}$ are calculated, which are used for the following search step. Each search step 32 or 33 can be seen as a filtering applied to the data matrix $\{R'\}$ to reduce the incidence of the noise samples B_{j_1, j_2} on the components R'_{j_1, j_2} of this matrix.

The steps 32 and 33 are essentially identical if the role of the rows and of the columns of the matrices are exchanged. On initialization 36 of the search step 32, the counting variable i is incremented by one unit, and the column index j_2 is initialized to 1. A decoding is performed, according to the code C_1 , of the data word corresponding to the j_2 -th column of the matrix $\{R'\}$ (step 37), which supplies new values of the components D_{j_1, j_2} and R'_{j_1, j_2} of the matrices $\{D\}$ and $\{R'\}$ ($1 \leq j \leq n_1$). The

decoding step 37 is followed by a comparison 38 between the column index j_2 and the number of columns n_2 . When j_2 remains lower than n_2 , the index j_2 is incremented by one unit (step 39), then the decoding step 37 is repeated.

5 When j_2 becomes equal to n_2 , all the columns having been processed, the other code word search step 33 of the decoding cycle in progress is commenced. On initialization 41 of the search step 33, the counting variable i is incremented by one unit, and the row index

10 j_1 is initialized to 1. Decoding is performed, according to the code C_2 , of the data word corresponding to the j_1 -th row of the matrix $\{R'\}$ (step 42), which supplies new values of the components $D_{j_1,j}$ and $R'_{j_1,j}$ of the matrices $\{D\}$ and $\{R'\}$. The decoding step 42 is followed by a

15 comparison 43 between the row index j_1 and the parameter n_1 of the code C_1 . When j_1 remains less than n_1 , the index j_1 is incremented by one unit (step 44), then the decoding step 42 is repeated. When j_1 becomes equal to n_1 , the code word search step 33 is terminated, and the counting

20 variable i is compared to $2m$ (test 45). When i remains less than $2m$, the search step 32 is reentered to commence the following decoding cycle. When i becomes equal to $2m$, the m decoding cycles having been accomplished, the $k_1 \times k_2$ decoded information bits \hat{a}_{j_1,j_2} are extracted from the

25 decision matrix $\{D\}$ produced during the last code word search step 33. With the systematic codes C_1 , C_2 applied in the way described above with reference to Figure 2, the \hat{a}_{j_1,j_2} 's can simply be recovered in the first k_1 rows and the first k_2 columns of the matrix $\{D\}$: $\hat{a}_{j_1,j_2} = D_{j_1,j_2}$

30 $(1 \leq j_1 \leq k_1, 1 \leq j_2 \leq k_2)$. These \hat{a}_{j_1,j_2} 's have the values -1 or +1; they can easily be converted to assume the values 0 or 1.

The step 37 of decoding of a data word corresponding to a column of the data matrix, in a first

35 implementation of the invention, is detailed on the flow chart of Figure 4. During this step 37, a data vector

[R'] and a decision vector [D], of length n_1 , are processed, respectively constituting subdivisions of the data matrix {R'} and of the decision matrix {D}: $R'_j = R'_{j,j2}$ and $D_j = D_{j,j2}$ ($1 \leq j \leq n_1$). First of all (step 51) the p least reliable components of the vector [R'], that is to say the components of [R'] which are closest to the binary decision threshold (zero), are marked. The indices corresponding to these p least reliable components are denoted r_1, r_2, \dots, r_p , with

$$|R'_{r1}| < |R'_j| \quad \forall j \neq r1$$

10

$$|R'_{r2}| < |R'_j| \quad \forall j \neq r1, r2$$

etc.

Having identified these p indices, q binary test sequences $[T^1], \dots, [T^q]$, of length n_1 , are constructed, then q binary words to be decoded $[U^1], \dots, [U^q]$ of length n_1 , by combining each of the q test sequences with the decision vector [D] (step 52). Each word $[U^s]$ is constructed in such a way that all its components other than those corresponding to the p indices r_1, \dots, r_p are equal to the corresponding components of the decision vector [D]: $U_j^s = D_j$ for $j \neq r_1, \dots, r_p$. It generally suffices to take into account words $[U^s]$ which have only one or two components different from the corresponding components of the vector [D]. All these words are taken into account when $q = p(p+1)/2$. By way of example, when $p = 6$ and $q = 21$, the sequences $[T^s]$ and $[U^s]$ ($1 \leq s \leq q$) can be constructed in the following way:

* the $p = 6$ first test sequences $[T^s]$ have one

bit equal to +1 in position rs and bits equal to -1 at the other positions: $T_{rs}^s = +1$ et $T_j^s = -1$ for $1 \leq s \leq 6$ and $j \neq rs$;

5

$$\begin{array}{ll}
 * \quad [T^7] = [T^1] \oplus [T^2] & [T^{15}] = [T^2] \oplus [T^6] \\
 [T^8] = [T^1] \oplus [T^3] & [T^{16}] = [T^3] \oplus [T^4] \\
 [T^9] = [T^1] \oplus [T^4] & [T^{17}] = [T^3] \oplus [T^5] \\
 [T^{10}] = [T^1] \oplus [T^5] & [T^{18}] = [T^3] \oplus [T^6] \\
 10 \quad [T^{11}] = [T^1] \oplus [T^6] & [T^{19}] = [T^4] \oplus [T^5] \\
 [T^{12}] = [T^2] \oplus [T^3] & [T^{20}] = [T^4] \oplus [T^6] \\
 [T^{13}] = [T^2] \oplus [T^4] & [T^{21}] = [T^5] \oplus [T^6] \\
 [T^{14}] = [T^2] \oplus [T^5] &
 \end{array}$$

15

where \oplus designates the exclusive OR operation, component by component, between two vectors:

$$* \quad [U^s] = [T^s] \oplus [D] \quad \text{for } 1 \leq s \leq q$$

20

At the following step 53, algebraic decoding of the decision vector $[D]$ and of the q words $[U^s]$ is performed. For this algebraic decoding, in the case of BCH codes, a Berlekamp decoder is used, for example, which is well known in the field of block coding (see E.R. Berlekamp, "Algebraic Coding Theory", McGraw-Hill, New York, 1968). The $q + 1$ elementary decodings supply q' code words $[C^1], \dots, [C^{q'}]$ of the code C_1 . In the general case, $q' \leq q + 1$, since, on the one hand, certain code words may appear several times in the decoding results, and, on the other hand, the algebraic decoder may not find certain code words if the signal is very distorted. The words supplied as results of the algebraic decoding thus have to be checked in order to determine whether they do or do not constitute words of the code C_1 . This checking can be performed simply by multiplying each word obtained

25

30

35

by the parity check matrix relative to the code C_1 , and by eliminating the word if the result of the multiplication is not zero. However, in the case in which the code C_1 is perfect (that is to say that no word of n_1 bits is spaced
 5 from all the possible code words by more than $(d_1-1)/2$, which is the case particularly for the Hamming codes), the step of checking the results from the algebraic decoder is pointless.

Among the q' code words found, the one $[C^d]$ which
 10 exhibits the smallest euclidean distance $M^d = \|[C^d] - [R']\|^2$ with the data vector $[R']$ is selected (step 54). This word $[C^d]$ will constitute the next decision vector. As candidate code word $[C^c]$ is also selected that which, among the q code words found except for the word $[C^d]$, has
 15 the shortest euclidean distance $M^c = \|[C^c] - [R']\|^2$ from the data vector $[R']$. This candidate word will be the only one which can be used as concurrent word for the calculation of confidences associated with the different bits of the word $[C^d]$.

20 Next a loop is performed for calculation from the components W_j of a correction vector $[W]$ ($1 \leq j \leq n_1$). At the beginning of this loop (step 55), the index of component index j is initialised to 1. At each iteration in this loop, a test step 56 is carried out to determine
 25 if the j -th component of the candidate word $[C^c]$ is different from that of the selected code word $[C^d]$ ($C_j^c \neq C_j^d$). If so, the candidate code word $[C^c]$ is a concurrent word with respect to the j -th component. The component W_j is then calculated at step 58 according to the
 30 formula:

$$W_j = \left(\frac{M^c - M^d}{4} - C_j^d R_j' \right) C_j^d$$

It will be observed that the quantity $M^c - M^d$ occurring in this formula is always positive, so that $|M^c - M^d| = M^c - M^d$. If the test step 56 reveals that $C_j^c = C_j^d$, i.e. if no concurrent word can be determined, the component W_j is calculated at step 59 according to the formula:

$$W_j = (\beta_i - C_j^d \cdot R_j) \cdot C_j^d$$

(2)

where β_i designates a positive confidence coefficient.

10 After calculation of the correction component W_j , the component index j is compared to the length n_1 of the vector $[R']$ (step 60). When j remains lower than n_1 , the index j is incremented by one unit (step 61), and the following iteration is performed, starting with the test 56.

15 When j becomes equal to n_1 , the loop is terminated, and the decoding step 37 terminates with the updating 62 of the data vector $[R']$ and of the decision vector $[D]$. The new vector $[R']$ is taken to be equal to the sum of the input vector $[R]$ (each component R_j of which is extracted from the input matrix $\{R\}$: $R_j = R_{j,12}$), and from the correction vector $[W]$ multiplied by another positive confidence coefficient α_i : $[R'] = [R] + \alpha_i[W]$.
20 The new decision vector $[D]$ is taken to be equal to the code word $[C^d]$ selected at step 54.

25 In an execution variant, the formula (2) applied where appropriate at step 59 is replaced by:

$$W_j = \beta_i C_j^d$$

(2')

which procures a correction W_j directly proportional to the sign of the new decision C_j^d . Other formulae making use of a confidence coefficient could also be used when no concurrent word is identified.

5 The steps 42 of decoding of data words corresponding to rows of the data matrix are similar to the steps 37 detailed above with reference to Figure 4, by replacing the code C_1 by the code C_2 , and the length n_1 by the length n_2 , and by dividing the matrices $\{R'\}$, $\{D\}$, $\{R\}$
10 not into column vectors $\{R'\}$, $\{D\}$, $\{R\}$, but into row vectors.

 The confidence coefficients α_i and β_i are assigned an index corresponding to the counting variable i shown in the flowchart in Figure 3. Indeed, these coefficients α_i ,
15 β_i can vary from one search step 32, 33 to another. The α_i and β_i values preferably increase as the code word search steps 32, 33 progress, to reflect the increasing reliability of the decoding.

 As an illustration of the performances of the
20 decoding process illustrated above, Figure 5 shows BER curves as a function of the signal to noise ratio E_b/N_0 obtained by simulation in the case of the product of two identical elementary codes BCH(64,57,4). In this case, the elementary decoder used to decode the lines and
25 columns (steps 37,42), necessitates approximately 43000 logic gates when decoding conforms to the optimum version described in EP-A-0 654 910. Among these 43000 gates, 25000 are used for calculation of the correction vector $[W]$, i.e. 59% of the circuit. With the elementary decoding
30 illustrated in Figure 4, the number of gates used for the calculation of $[W]$ is divided by 10. The elementary decoder can thus be implemented with approximately 20500 gates instead of 43000. The results in Figure 5 were obtained in the case of modulation by quadrature phase
35 shift keying (QPSK) and of an additive white Gaussian

noise channel, the data being quantified over 4 bits. $m=4$ decoding cycles were applied with $q=16$ test sequences built from the $p=4$ least weak components of the data vector $[R']$. During the $2m=8$ code word search steps, the successive values of the coefficient α_i were 0.4, 0.4, 0.5, 0.5, 0.6, 0.6, 0.65, 0.65, while the coefficient β_i remained constant: $\beta_i=7$. The curve II shows the results obtained by applying the optimum version of the decoding process according to EP-A-0 654 910. The curve III shows the corresponding results in the case of elementary decoding according to Figure 4. For comparison, curve I shows the performances observed in the absence of channel coding. It is observed that the degradation of performances brought about by the simplification of the elementary decoding remains less than 0.15 dB for a BER of 10^{-5} . This degradation is slight if weighed against the 50% gain in terms of circuit complexity. The implementation of a larger number of iterations can be envisaged, using the same elementary circuit, thus entailing an additional reduction in the overall complexity of the circuit.

Figure 6 shows an architecture of a BTC circuit suitable for carrying out the decoding of the product code according to an algorithm such as that described above, in the specific case where the elementary codes used are identical. Each of the elementary decoding steps according to Figure 4 is executed by a dedicated arithmetic and logic circuit 65 controlled by a control processor 66 on the BTC circuit (it would also be possible for several circuits 65 to be used to carry out several elementary decodings in parallel). A RAM memory 67 is used to store the samples of the matrices $\{R\}$, $\{R'\}$ and $\{D\}$.

The processor 66 supervises the decoding according to the general flowchart in Figure 3. On receipt of the

samples of the signal $R(t)$, the processor 66 orders writing to the memory 67 in order to form the input matrix $\{R\}$ (step 30) and store it at the appropriate addresses, and to build the matrices $\{R'\}$ and $\{D\}$ (step 31) and store them at the appropriate addresses. At each elementary decoding 37 or 42, the processor orders read operations from the memory 67 to supply the appropriate samples for the vectors $\{R'\}$, $\{D\}$ and $\{R\}$ to the elementary decoder 65, then write operations to record the new values for these
10 vectors $\{R'\}$ and $\{D\}$. At the end of the m cycles, the processor 66 executes step 46, ordering read operations at the appropriate addresses in the memory 67 (matrix $\{D\}$) in order to issue the output signal $Z(t)$ of the decoder 17.

The invention allows various product code
15 parameters to be varied while using the same BTC circuit 17 for the decoding: all that is required is for the appropriate parameters to be supplied to the part of the program of the processor 66 which relates to the formation of the input matrix $\{R\}$ at step 30, and optionally to the
20 part which relates to the extraction of the information bits at step 46.

Programming also occurs with respect to the coder 12, of which Figure 7 shows a block diagram in the specific case where the elementary codes used are
25 identical. A conventional arithmetical circuit 68 is used for successive elementary codings of the lines and columns of the matrix of bits to be transmitted (steps 22 and 23 of Figure 2). This elementary coder 68 is controlled by a control processor 69 of the coder. A RAM memory 70 is
30 used to store the samples of the matrix $\{c\}$.

On receipt of a block of binary samples of the signal $X(t)$ (here labelled a_u for $u=1,2,\dots,k-X$ where $k=k_1, k_2$), the processor 69 orders write operations in the memory 70 in order to form the matrix $\{a\}$ with k_1 lines
35 and k_2 columns, which is a sub-matrix of the matrix $\{c\}$ owing to the systematic coding (step 21). On each

elementary coding, the processor 69 orders read operations in the memory 70 to supply the coder 68 with the appropriate samples of the row or column to be coded, and write operations to record the values of the redundancy bits obtained. After the n_1+n_2 elementary codings, the bits of the final matrix {c} are available in the memory 70, and the processor 69 orders read operations at the appropriate addresses in this memory in order to supply the signal $Y(t)$ to the modulator. The binary samples of the signal $Y(t)$ are here labelled c_v for $v=1,2,\dots,n-X-Y$ where $n=n_1.n_2$.

Programming of the coder and the decoder allows the application of a shortening technique and/or a puncturing technique to the product code.

In the case of shortening, programming consists in supplying the coder and the decoder with the number X which represents the difference between the number of bits k of the matrix {a} to which the product code is applied and the number $k-X$ of bits a_v per block to be coded. From this number X , the coder determines X positions in the matrix {a} for bits of determined value(s) (for example 0) which, when each block is processed, will be located in corresponding positions of the matrix {c} and which will be excluded from the bits transmitted c_v . It also determines an order in which the bits a_v of each block will be stored in the other positions of the matrix {a}.

To determine these positions, the processors 69, 66 of the coder and the decoder apply a predefined procedure, such as for example the procedure the flowchart of which is shown in Figure 8. In this example, a matrix {h} with k_1 lines and k_2 columns designates, by $h_{i,j}=1$, the positions i,j of the known bits. Initially, all the components of the matrix {h} are at 0, as are the indexes i,j and n_x (step 80). The index n_x is compared with X at step 81, and if $n_x < X$, the indexes i and j are incremented by 1, respectively modulo k_1 and modulo k_2 , at step 82 (in

the notations used here, the indexes i and j are respectively included between 1 and k_1 and between 1 and k_2 such that the incrementation is carried out according to the formulae indicated in block 82 in Figure 8). If $h_{i,j} \neq 0$ after step 82 (test 83), the column index j is incremented by 1 modulo k_2 at step 84 before the test 83 is repeated.

When the test 83 shows that $h_{i,j} = 0$, the value 1 is assigned to this component $h_{i,j}$ at step 85, and the index nx is incremented by 1 before returning to the comparison 81. All the positions are assigned when the comparison 81 shows that $nx = X$.

The above procedure allows uniform distribution of the positions of the known bits on the lines and the columns of the matrix $\{a\}$. The uniformity is perfect, that is, all the lines have the same number of $h_{i,j} = 1$ positions as do all the columns, when X is a multiple of k_1 and of k_2 ; otherwise, deviations from uniformity are minimised. Figure 9 illustrates one form of the matrix $\{h\}$ in the specific case where $k_1 = k_2 = 10$ and $X = 30$ (empty boxes corresponding to $h_{i,j} = 0$).

After determining the X positions where $h_{i,j} = 1$, the processors 69, 66 each calculate two tables $x(u)$, $y(u)$ ($1 \leq u \leq X$) respectively providing the row indexes and the column indexes for the positions of the matrix $\{a\}$ where the successive bits a_u of each block to be coded will be stored. These tables are obtained at step 86 by assigning in a determined order the positions i, j of the matrix $\{a\}$ such as $h_{i,j} = 1$, for example row by row ($i = 1, 2, \dots, k_1$) and, in each row in ascending order of the column indexes ($j = 1, 2, \dots, k_2$).

The procedure of Figure 8 is carried out once on programming of the coder and the decoder, the tables $x(u)$ and $y(u)$ and the matrix $\{h\}$ then being retained in memory.

For each block of bits a_u , the processor 69 of the coder 12 will build the matrix $\{a\}$ at step 21 according to:

$$a_{i,j}=0 \quad \text{if} \quad h_{i,j}=1$$

$$a_{x(u),y(u)}=a_u \quad \text{for} \quad 1 \leq u \leq k-X$$

In its output signal corresponding to a block, the coder 12 does not include the bits $c_{i,j}$ such as $h_{i,j}=1$ (step 24). In building the matrix $\{R\}$ at step 30, the processor 66 of the decoder 17 places in these positions samples $R_{i,j}$ the sign of which corresponds to the known value of the bit $a_{i,j}$ (for example -1 for $a_{i,j}=0$), and the absolute value M of which represents maximum confidence (typically the greatest of the quantification values of the decoder).

At the end of block decoding (step 46), the processor 66 extracts the estimations \hat{a}_u ($=\pm 1$) of the bits a_u , according to $\hat{a}_u = D_{x(u),y(u)}$.

As regards puncturing, programming consists in supplying the coder and the decoder with the number Y which represents the difference between the number $n-X$ of unknown bits of the matrix $\{c\}$ resulting from the application of the product code ($X=0$ if no code shortening is carried out) and the number of bits $n-X-Y$ transmitted by the coder for each information block. From this number Y , the coder determines Y positions in the matrix $\{c\}$ for bits which will be excluded from the transmitted bits c_v .

To determine these positions, the processors 69, 66 of the coder and the decoder apply a predefined procedure, such as for example the procedure the flowchart of which is shown in Figure 10, which is similar to that in Figure 8. In this example, the dimensions of the matrix $\{h\}$ have been extended to n_1 lines and n_2 columns; and the Y positions i,j of the punctured bits are designated by $h_{i,j}=2$. Initially, all the components of the matrix $\{h\}$ are at 0, except those set to 1 by the procedure in Figure 8 if $X \neq 0$, and the indexes i,j and ny (step 100). The index ny is compared with Y at step 101, and if $ny < Y$, the indexes i and j are incremented by 1, respectively modulo n_1 and modulo n_2 , at step 102 (in the notations used here, the indexes i and j are respectively

comprised between 1 and n_1 and between 1 and n_2 such that the incrementation is carried out according to the formulae indicated in block 102 in Figure 10). If $h_{i,j} \neq 0$ after step 102 (test 103), the column index j is incremented by 1 modulo n_2 at step 104 before the test 103 is repeated. When the test 103 shows that $h_{i,j} = 0$, the value 2 is assigned to this component $h_{i,j}$ at step 105, and the index nx is incremented by 1 before returning to the comparison 101. All the positions are assigned when the comparison 101 shows that $ny=Y$.

The above procedure allows uniform distribution of the Y positions of the punctured bits on the lines and the columns of the matrix $\{c\}$. If there is no shortening, the uniformity is perfect when Y is a multiple of n_1 and n_2 ; otherwise, variations from uniformity are minimised. Figure 11 illustrates one form of the matrix $\{h\}$ in the case where $n_1=n_2=12$ and $Y=24$ without shortening (empty boxes corresponding to $h_{i,j}=0$).

When shortening is used in conjunction with puncturing ($X \neq 0$ and $Y \neq 0$), the uniformity of the Y positions on the lines and columns of the matrix $\{c\}$ is perfect in the case of square matrices ($k_1=k_2$ and $n_1=n_2$) if Y is a multiple of n_1 ; otherwise, the variations in uniformity remain very slight. Figure 12 illustrates one form of the matrix $\{h\}$ in the case where $k_1=k_2=10$, $n_1=n_2=12$, $X=30$ and $Y=24$. Only the $n-X-Y=90$ bits $c_{i,j}$ placed in the empty boxes in Figure 12 are transmitted by the coder.

Transmission of the $n-X-Y$ bits c_v is carried out in a specific order, for example row by row, step 24 then consisting, for the coder, in carrying out:

$$C_v = C_{x'(v), y'(v)} \quad \text{for} \quad 1 \leq v \leq n-X-Y,$$

the row and column indexes $x'(v)$, $y'(v)$ being determined and memorised on programming of the coder and the decoder, at step 106 (Figure 10). Correspondingly, the processor 66 of the decoder 17 places the $n-X-Y$ samples R_v of the

received signal block in the appropriate locations in the matrix $\{R\}$ at step 30:

$$R_{x'(v),y'(v)} = R_v \quad \text{for } 1 \leq v \leq n-X-Y.$$

In the other locations, the processor 66 places:

- 5 . $R_{i,j} = -M$ if $h_{i,j} = 1$ as indicated previously,
- . $R_{i,j} = \pm \epsilon$ if $h_{i,j} = 2$

The number ϵ represents a maximum confidence in the estimations of the Y_s (typically the lowest of the decoder quantification values).

- 10 During the iterative decoding, the estimations $D_{i,j}$ of the bits such as $h_{i,j} = 1$ will remain very low. Those relating to the punctured bits ($h_{i,j} = 2$) will see their reliabilities increase as the elementary decodings progress.

- 15 The performances of the shortening and puncturing techniques applied to a product code according to the invention are respectively illustrated by the Figures 13 and 14 in the case of the product of two identical elementary codes BCH (32,26,4), and of a modulation by
- 20 quadrature phase shift keying (QPSK). The parameters of the product code are then:

- . $k=676$
- . $n=1024$
- . $d=16$
- 25 . initial efficiency 0.660.

- In Figures 13 and 14, the curve I shows the evolution of the binary error rate (BER) as a function of the signal/noise ratio E_b/N_0 in the absence of channel coding, and the curve IV with the product code applied
- 30 without shortening or puncturing ($X=Y=0$) and with an iterative decoding in $m=4$ cycles.

- Figure 13 corresponds to the case where $X=312$, $Y=0$, that is, each block to be coded comprises $k-X=364$ information bits and the coding efficiency is $(k-X)/(n-X)=0.511$, the theoretical Shannon limit then being 0.04 dB. The curves V.1 to V.4 show the performances obtained
- 35

after, respectively, $m=1$, $m=2$, $m=3$ and $m=4$ decoding cycles. It is observed that the gradient of the curve after 4 cycles is practically unchanged with respect to the case where k information bits are transmitted per block (curve IV). The signal to noise ratio needed to obtain a $BER=10^{-5}$ after 4 cycles is equal to 2.91 dB, which corresponds to approximately 2.9 dB above the Shannon limit. This solution makes a BTC circuit available which can transmit data blocks of any size, less than or equal to k , without any significant degradation of codec performances with respect to the theoretical Shannon limit.

Figure 14 corresponds to the case where the same BTC circuit is programmed with $X=0$, $Y=104$, that is, each coded block comprises $n-k-Y=244$ redundancy bits instead of 348, and the coding efficiency is $k/(n-Y)=0.735$, the theoretical Shannon limit then being 0.81 dB. The curves VI.1 to VI.4 show the performances obtained after, respectively, $m=1$, $m=2$, $m=3$ and $m=4$ decoding cycles, in the case where the punctured bits are selected from the redundancy bits added by the systematic product code. It is observed that the gradient of the curve after 4 cycles is practically unchanged with respect to the case where the n bits of the matrix are transmitted (curve IV). The signal to noise ratio needed to obtain a $BER=10^{-5}$ after 4 cycles is equal to 3.71 dB, which corresponds to approximately 2.9 dB above the Shannon limit. This solution makes a BTC circuit available where any number of redundancy bits are used, less than or equal to $n-k$, without any significant degradation of codec performance with respect to the theoretical Shannon limit.

「以下 余白」

4 B r i e f D e s c r i p t i o n o f D r a w i n g s

- Figure 1 is a block diagram of a digital transmission chain which can be used to implement the process according to the invention;

- Figure 2 is a flowchart illustrating the application of a product code;

- Figure 3 is a general flowchart of an iterative decoding phase which can be used according to the invention;

- Figure 4 is a flowchart detailing an elementary decoding step for a row or a column;

- Figure 5 is a graph illustrating the performance of the iterative decoding according to Figures 3 and 4;

- Figures 6 and 7 are respective block diagrams of a channel decoding circuit and a channel coding circuit according to the invention;

- Figure 8 is a flowchart of an allocation procedure which can be used for the shortening of a product code;

- Figure 9 is a diagram illustrating the results of the procedure in Figure 8;

- Figure 10 is a flowchart of an allocation procedure which can be used for the puncturing of a product code;

- Figures 11 and 12 are diagrams illustrating the results of the procedure in Figure 10; and

- Figures 13 and 14 are graphs illustrating the performances of the iterative decoding applied to a shortened product code and to a punctured product code.

FIG. 1

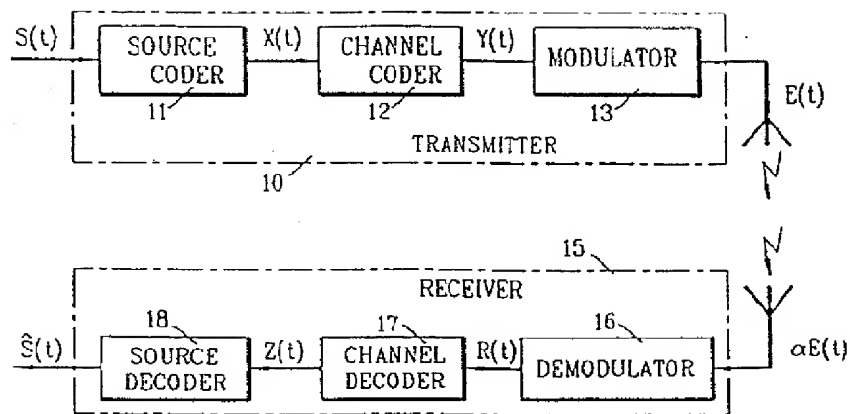


FIG. 2

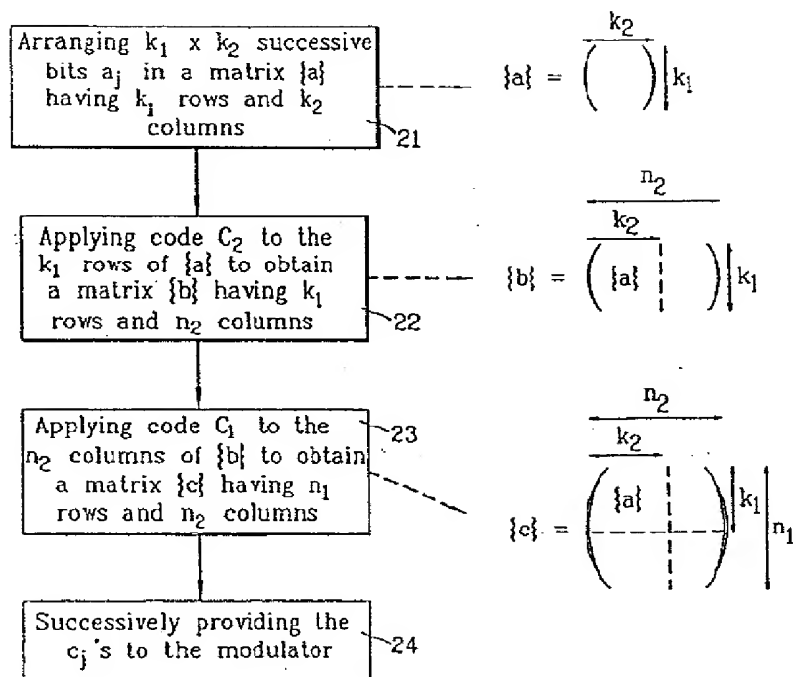
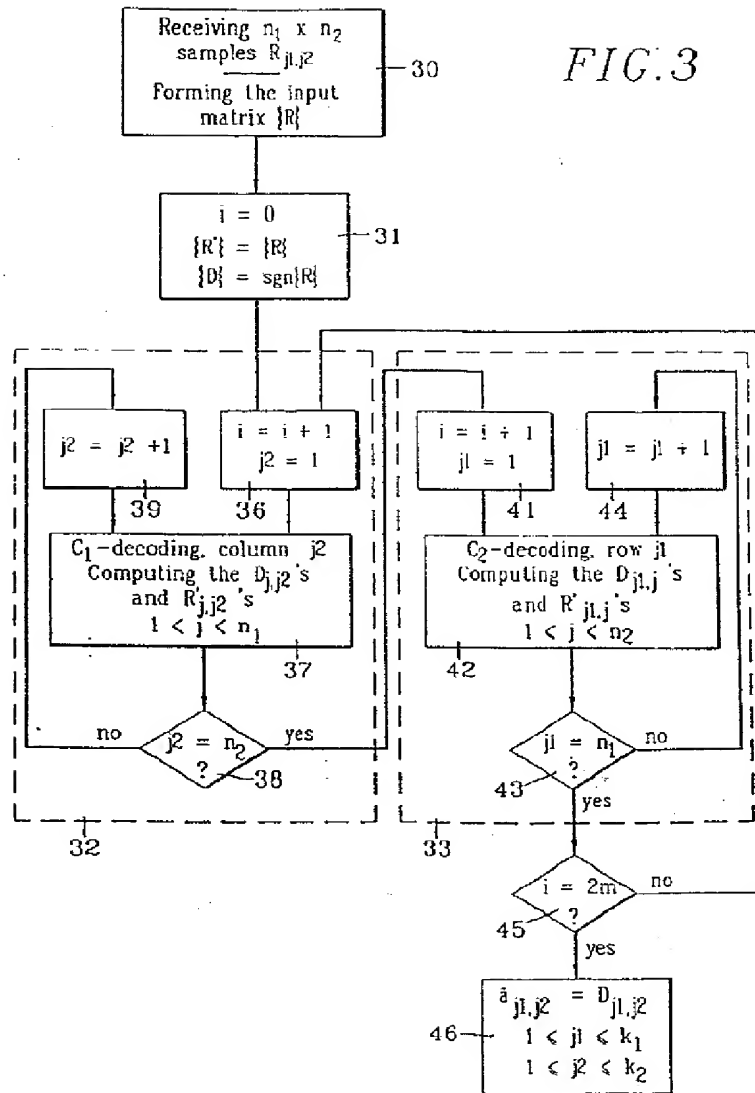
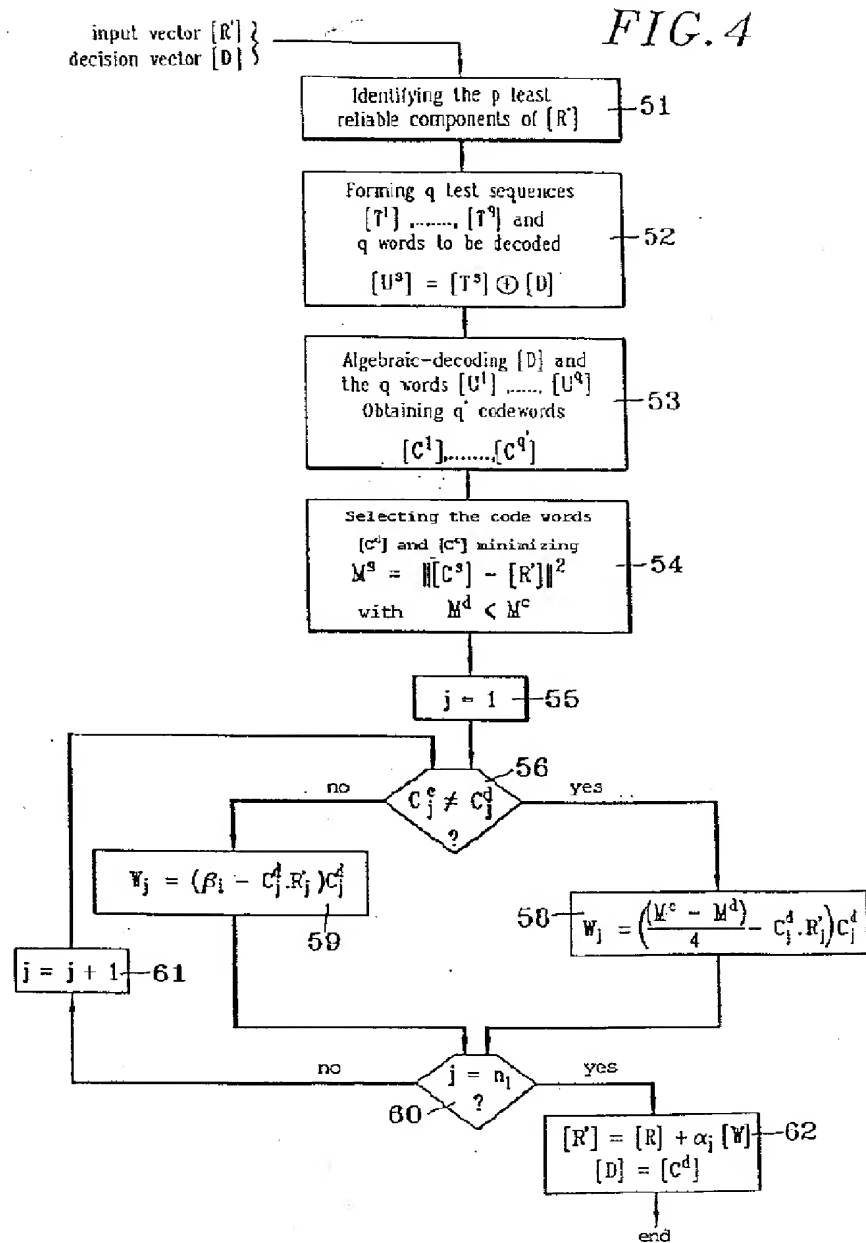


FIG. 3





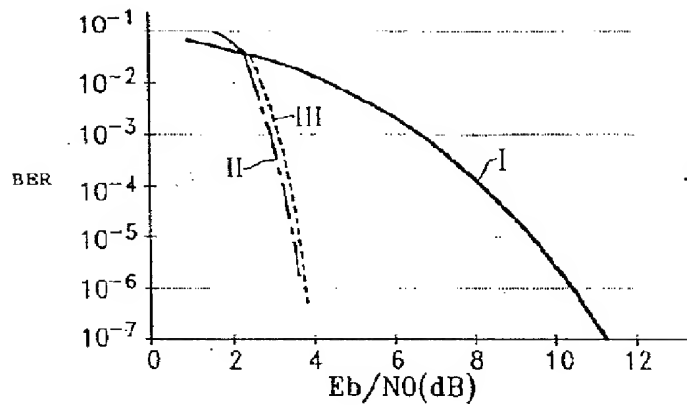


FIG. 5

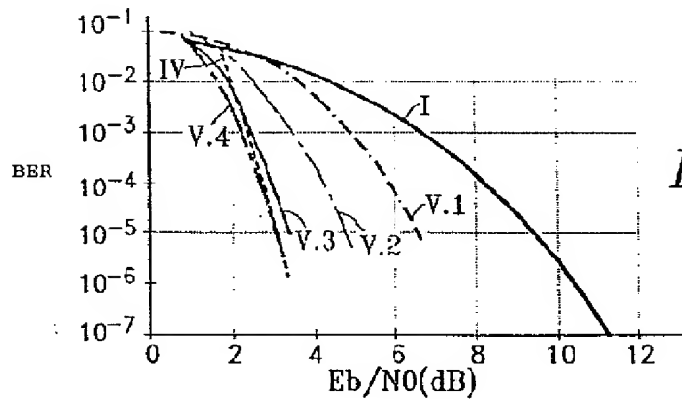


FIG. 13

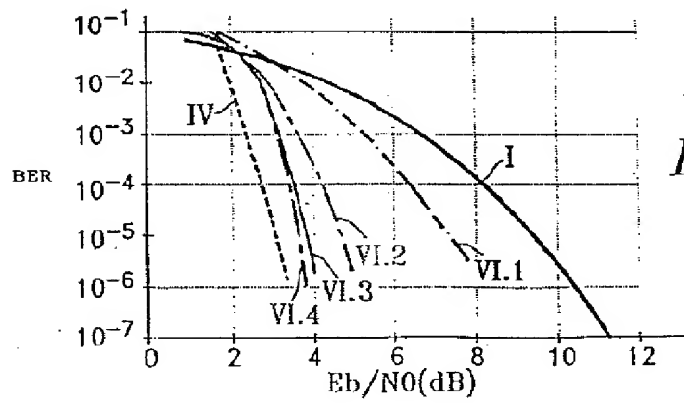


FIG. 14

FIG. 6

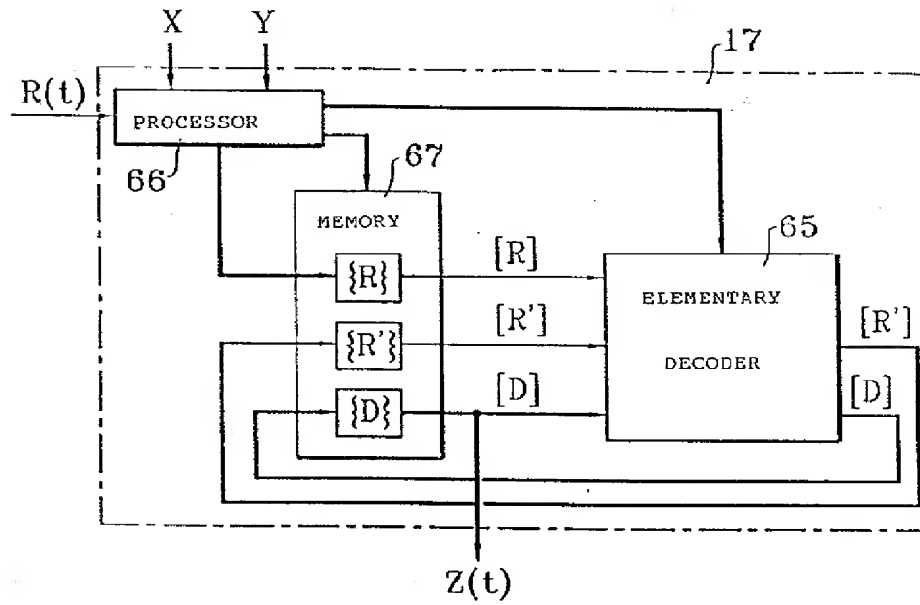


FIG. 7

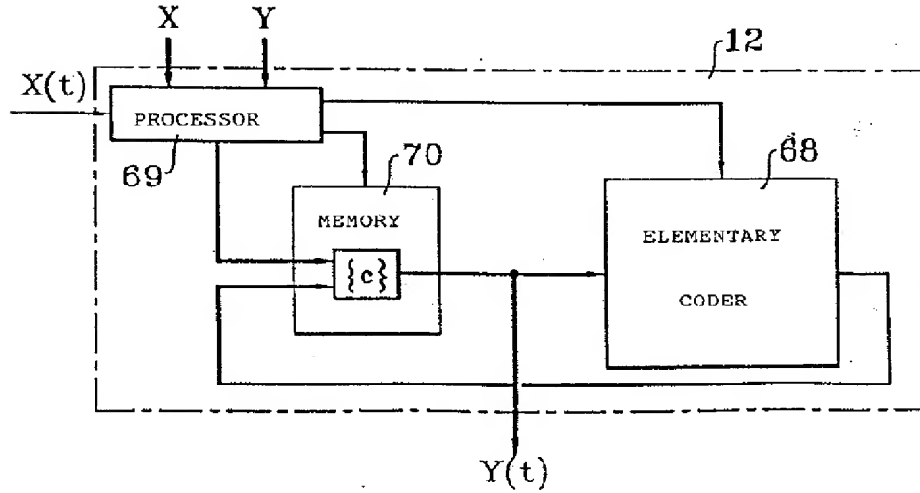


FIG. 8

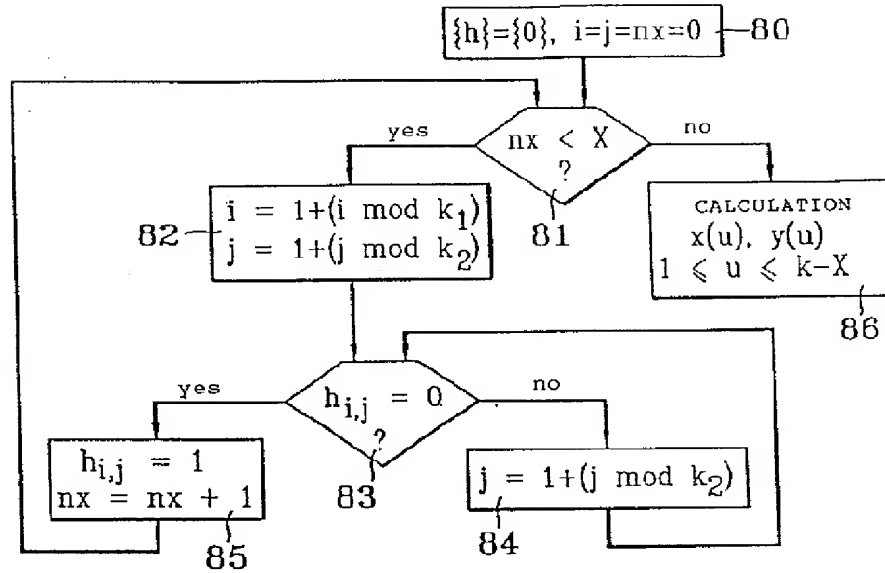


FIG. 9

		j									
		1	2	---	---	---	---	---	---	k ₂	
i	1	1	1	1							
	2		1	1	1						
				1	1	1					
					1	1	1				
						1	1	1			
							1	1	1		
								1	1	1	
									1	1	1
										1	1
	k ₁	1	1								1

FIG. 10

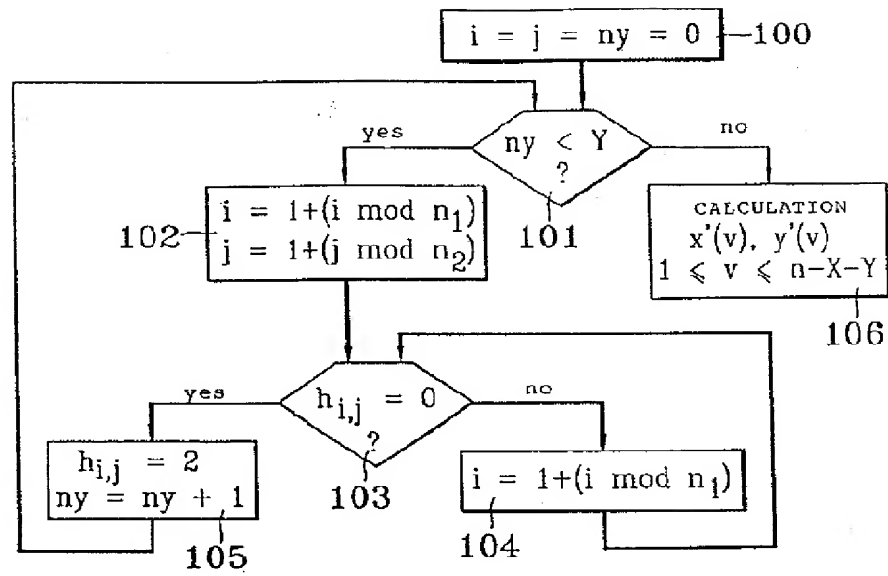


FIG. 11

		j																
		1	2	-----										n_2				
i	1	2															2	
	2	2	2															
			2	2														
				2	2													
					2	2												
						2	2											
							2	2										
								2	2									
									2	2								
										2	2							
											2	2						
	n_1												2	2			2	2

j

[illegible]

1 Abstract

The bits transmitted are coded according to the product of at least two systematic block codes. Iterative decoding is applied in order to determine, at each code word search step, a data matrix ($\{R\}$) and a decision matrix ($\{D\}$) used for the following step. The new decision matrix is determined at each step by decoding the lines or columns of the input matrix, and the new data matrix is determined taking into account the correction terms which increase the reliability of the decoding on each iteration. The coding and decoding circuits (17) are rendered programmable by a shortening technique allowing selection of the number $k-X$ of non-redundant information bits per block to be coded. Known values are assigned to the other bits, the positions of which are uniformly distributed according to each dimension of the matrices.

2 Representative Drawing

Fig. 6